UNISYS

CTOS™ Generic Print System™

Using the Font Tool

Release

Priced Item

April 1990
Distribution Code S
Printed in USA
09–02449

CTOS™ Generic Print System™

Using the Font Tool

Copyright © 1990 Unisys Corporation
All Rights Reserved
Unisys is a trademark of Unisys Corporation

Release: April 1990
Distribution Code S
Printed in USA

Priced Item: 09–02449

The names, places, and/or events used in this publication are not intended to correspond to any individual, group, or association existing, living, or otherwise. Any similarity or likeness of the names, places, and/or events with the names of any individual, living or otherwise, or that of any group or association is purely coincidental and unintentional.

NO WARRANTIES OF ANY NATURE ARE EXTENDED BY THIS DOCUMENT. Any product and related material disclosed herein are only furnished pursuant and subject to the terms and conditions of a duly executed Program Product License or Agreement to purchase or lease equipment. The only warranties made by Unisys, if any, with respect to the products described in this document are set forth in such License or Agreement. Unisys cannot accept any financial or other responsibility that may be the result of your use of the information or software material, including direct, indirect, special or consequential damages.

You should be careful to ensure that the use of this information and/or software material complies with the laws, rules, and regulations of the jurisdictions with respect to which it is used.

The information contained herein is subject to change without notice. Revisions may be issued to advise of such changes and/or additions.

Convergent, Convergent Technologies, NGEN, and WGS/SpreadSheet are registered trademarks of Convergent, Inc.

OFIS is a registered trademark of Unisys Corporation.

Art Designer, AutoBoot, ClusterCard, ClusterShare, Context Manager, Context Manager/VM, CT-DBMS, CT-MAIL, CT-Net, CTOS, CTOS/VM, CWS, Document Designer, Generic Print System, Image Designer, Network PC, PC Emulator, Phone Memo Manager, Print Manager, Series 286i, Series 386i, Shared Resource Processor, Solution Designer, SRP, TeleCluster, The Cluster, The Operator, Voice/Data Services, Voice Processor, and X-Bus are trademarks of Convergent, Inc.

Diablo is a registered trademark of Xerox Corp. Epson is a registered trademark of Epson America, Inc. Hewlett-Packard and LaserJet are registered trademarks of Hewlett-Packard Corp. LaserWriter is a registered trademark of Apple Computer, Inc. MS-DOS is a registered trademark of Microsoft Corp. NEC is a registered trademark of Nippon Electric Corporation Information Systems. PostScript is a registered trademark of Adobe Systems, Inc. Qume Sprint is a registered trademark of Qume Corp. Texas Instruments is a registered trademark of Texas Instruments, Inc. Imagen is a trademark of Imagen Corp. UNIX is a registered trademark of AT&T.

Page Status

Page	Issue
iii	Original
iv	Blank
v–xiii	Original
xiv	Blank
xv–xvi	Original
1–1 through 1–11	Original
1–12	Blank
2–1 through 2–52	Original
3–1 through 3–54	Original
4–1 through 4–22	Original
5–1 through 5–22	Original
A–1 through A–20	Original
B–1 through B–5	Original
B–6	Blank
C–1 through C–3	Original
C–4	Blank
D–1 through D–5	Original
D–6	Blank
E–1 through E–4	Original
F–1 through F–4	Original
G–1 through G–8	Original
Glossary–1 through Glossary–9	Original
Glossary–10	Blank
Index–1 through Index–12	Original

Contents

About This Manual

Who Should Use This Manual xv
How This Manual Is Organized xv
What's New in the Font Tool? xvi
Where to Go for More Information xvi

Section 1. Introduction

What Is the Font Tool? 1–1
When Do You Need the Font Tool.? 1–1
What You Can't Do With the Font Tool 1–2
Font Terminology 1–3
The Font Database 1–4
The Font Service 1–5
Applications and the Font Service 1–6
Editing the Font Database 1–8
How the Components of the Font Database Work Together 1–10

Section 2. Inside the Font Tool

Organization of the Editable Font Database 2–1
Variant Font Databases 2–2
Keeping Track of Changes 2–5
Preventing Changes From Being Overwritten 2–5
Recovering Original Data 2–6
Minimizing Redundant Font Data 2–7
When to Use a Variant Database 2–7
The Main Menu 2–10
Font Families 2–11
When to Use the Font Families Directory 2–12
The Default Font Family 2–12
Similar Font Families 2–13

Device Types 2–13
When to Use the Device Types Directory 2–14
The Default Device Type 2–15
Downloadable Fonts 2–15
Device Types in the Standard Editable Font Database . 2–16
Translation Tables 2–17
When to Add or Modify Translation Tables 2–19
The Translation Tables Directory 2–20
Sample First Level Translation Table 2–21
Sample Second Level Translation Table 2–23
Translation Table Escapes 2–24
Character Sets 2–25
When to Add or Modify Character Sets 2–25
The Character Sets Directory 2–26
Sample Character Set Detail Form 2–26
Width Tables 2–27
When to Add Width Tables 2–28
The Width Tables Directory 2–28
Sample Width Table 2–29
Font Keys 2–30
When to Add or Modify Font Keys 2–31
The Font Keys Directory and Detail Form 2–31
Alternate Character Sets 2–34
Raster Fonts 2–36
When to Modify Raster Fonts 2–37
The Raster Fonts Directory and Detail Form 2–37
Pseudo Font Families 2–39
When to Add Pseudo Font Families 2–40
The Pseudo Font Families Directory and Detail Form .. 2–40
Reduction Rules 2–41
When to Add Reduction Rules 2–42
Sample Reduction Rules 2–43
How Reduction Rules Are Created 2–46
Proper Rule Order 2–47
Nil Values 2–49
Font Tool Commands 2–49
Generate Runtime Database 2–50
Convert Wheel Set File 2–50
Import and Export 2–50
Batch Mode Commands 2–51
How Applications Use Font Data 2–51
Scenario 2–51

Section 3. Step-by-Step

Getting Ready 3–1
Guidelines for Working With the Font Tool 3–1
Getting Around in the Font Tool 3–2
Edit Mode and View Mode 3–3
Special Keys 3–3
Status Flags 3–6
Creating a Variant Database 3–7
Entering the Font Tool 3–8
The Main Menu 3–9
Working With Font Families 3–10
Modifying a Font Family Record 3–10
Adding a Font Family Record 3–12
Working With Device Types 3–13
Modifying a Record in the Device Types Directory 3–13
Changing the Default Device Type 3–14
Adding a Device Type to the Device Types Directory ... 3–14
Working With Translation Tables 3–17
Modifying a Translation Table 3–18
Creating a New Translation Table 3–19
Adding a Print Wheel Without Vendor Data 3–22
Invert and Shuffle Commands 3–23
Working With Character Sets 3–23
Creating a New Character Set 3–24
Working With Width Tables 3–26
Resize, Propagate, and Shuffle Commands 3–28
Working With Font Keys 3–29
Creating a New Font Key 3–29
Modify a Font Key 3–31
Tuning Raster Fonts 3–32
Working With Reduction Rules 3–39
Adding or Modifying a Reduction Rule 3–40
Testing Reduction Rules 3–40
Generating a New Runtime Font Database 3–41
Converting Wheel Sets 3–43
Exporting and Importing Data 3–47
Using the Export Command 3–48
Using the Export Variant Command 3–49
Using the Import Commands 3–50
Batch Mode Commands 3–51
Using Batch Mode Commands 3–54

Section 4. Using the Scaling Font Service

Overview 4–1
Components of the Scaling Font Service 4–2
Obtaining Prescaled Fonts 4–3
Configuring the Scaling Font Service 4–7
Font Nationalization 4–9
Character Codes 4–9
Description 4–9
Class 4–10
CTOS 4–10
BitStream Levels 1 and 2 4–11
PostScript Levels 1 and 2 4–12
Modifying Width Tables After Nationalization 4–18
Nationalize Scaling Fonts Utility 4–20
Nationalization of Utilities 4–22
Procedural Interface Changes 4–22

Section 5. Examples of Font Tool Tasks

Excluding Unused Font Data 5–1
Adding PostScript Font Families 5–3
Modifying HP LaserJet Font Keys 5–4
Adding an HP LaserJet Font Cartridge 5–6
Adding HP LaserJet Downloadable Fonts 5–9
Adding the HP LaserJet Linedraw Font 5–10
Exporting and Importing Data 5–14
Nationalizing Translation Tables 5–15

Appendix A CTOS and Screenfont Character Code Sets A–1

Appendix B Aliases B–1

Appendix C Decimal/Hexadecimal Conversion Table C–1

Appendix D Translation Tables and Character Sets D–1

Appendix E Translation Table Excapes E–1

Appendix F Font Database Character Set Conventions F–1

Appendix G Editable Font Database Format G–1

Glossary 1

Index 1

Figures

1–1. Screenfont Database 1–4
1–2. Document Designer Format Status Line 1–6
1–3. Document Designer, OFIS Designer, and the Font Service 1–7
1–4. Adding a Font Family to the Runtime Font Database 1–9
1–5. Removing a Font Family From the Runtime Font Database 1–10
1–6. Overview of Font Database 1–11

2–1. A Directory and a Detail Form 2–2
2–2. Variant and Original Databases 2–3
2–3. Adding to the Variant Database 2–4
2–4. Runtime Database Generated From the Original Editable Database . 2–5
2–5. Upgrading the Editable Font Database 2–6
2–6. Samples of Two Variant Databases 2–8
2–7. The Main Menu 2–10
2–8. Font Families Directory 2–11
2–9. Device Types Directory 2–14
2–10. CT Character Code Set (00h–7fh) 2–18
2–11. Translation Tables Directory 2–20
2–12. Sample First Level Translation Table 2–22
2–13. Sample Second Level Translation Table 2–24
2–14. Character Sets Directory and Detail Form 2–26
2–15. Width Table Directory 2–28
2–16. Sample Width Table 2–29
2–17. Font Key Data 2–30
2–18. Font Keys Directory 2–32
2–19. Sample Font Key 2–33
2–20. Primary and Alternate Character Sets: A Three Dimensional View .. 2–35
2–21. Primary and First Alternate Character Sets 2–36
2–22. Raster Fonts Directory 2–38
2–23. Raster Fonts Detail Form 2–39
2–24. Sample Pseudo Font Families Directory and Detail Form 2–40
2–25. The Reduction Table 2–43
2–26. Sample Reduction Rule 2–44

2–27. Sample Reduction Rule with Wild Cards 2–45
2–28. Two Reduction Rule Orders 2–48
2–29. Nil Value Reduction Rule 2–49

3–1. The Main Menu 3–9
3–2. Font Families Directory 3–11
3–3. Device Types Directory 3–13
3–4. Translation Tables Directory 3–18
3–5. Blank Translation Table 3–19
3–6. Character Codes in Translation Table 3–21
3–7. Blank Character Set Detail Form 3–24
3–8. Blank Width Table 3–27
3–9. Blank Font Key Detail Form 3–29
3–10. Raster Font Elements 3–33
3–11. Font Raster Detail Form 3–34
3–12. The Script Form 3–36
3–13. The Reduction Table 3–39
3–14. Wheel Set Conversion Form 3–44

5–1. Modified HP LaserJet Font Keys 5–5
5–2. HpTHelvBold14 Width Table 5–7
5–3. Font Key Detail Form for HP 14 Point Helvetica Bold 5–8
5–4. Linedraw Translation Table 5–12
5–5. HPSymbols Character Set 5–12
5–6. Linedraw Character Set in Courier Font Keys 5–13
5–7. CtToAndora Translation Table 5–18
5–8. Shuffle Table 5–19
5–9. Shuffling the Translation Tables 5–20
5–10. Completed Translation Table 5–21

Tables

4–1. Characters Available for Nationalization 4–13

A–1. Graphic Representation of the Standard Character Set A–2
A–2. Standard Character Set A–4
A–3. Screenfont Database Character Code Set A–14

C–1. Decimal/Hexadecimal Conversion Table C–2

About This Manual

This manual describes the Font Tool, what it is used for, and how to use it. The manual does not describe how to install the Font Tool. For installation procedures, see the Generic Print System 2.4 Release Notice.

Who Should Use This Manual

This manual is meant to be used by system administrators. If you are a system administrator, welcome to the Font Tool. If you aren't a system administrator, and you don't have one in your working environment, you can still use this manual, but you need to be familiar with your workstation.

How This Manual Is Organized

Section 1, "Introduction," presents an overview of the Font Database and the Font Tool. If you are new to the Font Database, Font Tool, or the Scaling Font Service, read this section first.

Section 2, "Inside the Font Tool," describes each part of the Font Tool in detail.

Section 3, "Step-by-Step," gives instructions on how to use each part of the Font Tool. If you go straight to this section without reading Section 2, you may run into a few unfamiliar terms and concepts.

Section 4, "Using the Scaling Font Service," details the Scaling Font Service, an alternative to the Font Service.

Section 5, "Examples of Font Tool Tasks," gives examples of the most common Font Tool tasks.

In addition to the five sections, there are several appendixes that provide additional reference material.

What's New in the Font Tool?

This manual describes the Font Tool as released with the Generic Print System 2.4. There is one primary change to the Font Tool in this release; support for the Scaling Font Service. The Scaling Font Service can be used instead of the Font Service. In addition to providing all the same tools as the Font Service, the Scaling Font Service gives you even more fonts and font sizes. It also gives you the ability to download bit-map fonts to printers that use the HP LaserJet device driver.

Where to Go for More Information

This manual is one of a three-part set of CTOS Generic Print System manuals. The other two manuals in this set are:

CTOS Generic Print System Using the Print Manager

CTOS Generic Print System Administration Guide

The documents listed below provide additional information related to this manual.

1.4 Screenfont Database Release Notice

2.4 Generic Print Service Release Notice

CTOS Office Applications Programmer's Guide

CTOS Editor User's Guide

The Executive manuals

Document Designer: System Administration

Getting Started With Your Workstation

Raster Font and Icon Designer manuals

Section 1
Introduction

In this age of office publishing, a professional look in business documents and publications has become standard. This requires some rather sophisticated software. This book describes one aspect of that software, the Font Tool.

What Is the Font Tool?

When you use application software such as Document Designer and OFIS Designer, you want to be able to print every font that your printer can print. This is only possible if your system has the correct *font data*. Font data is used by your application software to lay out text correctly, to make sure the correct characters are printed, and to inform the printer which font cartridge or print wheel to use. The Font Tool is an application program that you use to add or modify font data. By doing so, you can take full advantage of your printer's capabilities.

When Do You Need the Font Tool?

The standard Font Database already supports a variety of printers and fonts. In most cases you can just install the Generic Print System and get right to work without ever using the Font Tool. There are certain situations, however, where you need to use the Font Tool, as described below:

- When your printer includes font families that aren't on your Document Designer or OFIS Designer *Fonts* menu.
- When you add a printer that is not supported by the Font Database. (For information on supported printers, see "Device Types in the Standard Editable Font Database," in Section 2.)

- When you add unsupported HP LaserJet font cartridges. Only the 92286A and 92286B font cartridges are supported by the standard runtime font database.
- When you want to print characters that are resident in your printer but are not included in the Font Database. These characters vary from printer to printer, but they are usually accented characters, such as E umlaut, or dingbat characters, such as hearts and stars.
- When you have documents that are formatted with custom wheel set files. If you have documents formatted with Word Processor, Document Designer 1.0, or OFIS Designer 1.0 software, you need to convert the wheel set files into data that can be used by Document Designer 2.0 or OFIS Designer 2.0, or higher. The wheel set files distributed with the Word Processor have already been converted and are included in the runtime font database. They are Courier 10, Courier 72, Cubic PS 96, Elite 12, and Scientific 10.
- When you want to distribute a customized Font Database. For example, if everyone in your company uses the same printers, you would want everyone to have the same font capabilities.
- When you need to nationalize the CTOS character code set.

What You Can't Do With the Font Tool

Although you can use the Font Tool to accomplish many tasks, there are certain things you can't do with it, which are described below.

- You can't use the Font Tool to create new fonts that your printer does not have. New fonts only come with new font cartridges, new print wheels, downloadable (soft) font files, or new printers.
- You can't use the Font Tool to edit existing fonts. For example, you can't make letters wider, or add customized typographical accents.
- You can't use the Font Tool to create new raster fonts or edit the bit map of existing raster fonts. This is done with the Raster Font Designer software.

Font Terminology

When working with fonts, you need to know the following terms. Most of them will be familiar to you if you have used Document Designer or OFIS Designer.

Points	Points are used to measure type height. A point is 1/72 of an inch. The type you are reading is 10 point type.
Type style	Bold, italic, bold italic, and roman (normal) are all type styles. You are reading roman type.
Font family	The overall design of the type. Type can be chunky, graceful, flashy, and so forth. Font families have names, for example, Helvetica, Avant Garde, P. T. Barnum, Dolores Marie, and so forth. The font family you are reading is New Century Schoolbook.
Font	A font describes all three of the above characteristics. *Times* is not a font; it's a font family. *Times bold* is not a font; it's one type style of the Times font family. *10 point Times bold* is a font; it describes the point size, type style, and font family. These definitions are often blurred in usage, for example, "Courier is my favorite font," or "Please specify the font and type size."
Character	Roughly synonymous with *letter*, but it also includes numbers and characters like ↑ and ".
Monospaced font family	A font family in which each character is the same width, for example, Courier.
Proportional font family	A font family in which characters have different widths, for example, Times or Helvetica.
Pitch	Used to measure the width of monospaced type. For example, 8 pitch means you can fit 8 letters in one inch.
Serif and Sans-serif	Serifs are the short lines that adorn the characters on some font families. Serif font families have serifs; sans-serif font families don't, as shown below.

Serif Sans-serif

The Font Database

The rest of this chapter explains how font data is used by your application software and by your device drivers (the programs that control your printers). When you understand how font data is used, it will be easier for you to understand how to use the Font Tool. Let's start with the Font Database.

The *Font Database* is a set of programs, tools, and files that enable you to print all the fonts that your printer is capable of printing. The Font Database can be divided into the following three parts:

Screenfont Database

The Screenfont Database is a *superset* of the Font Database. It contains everything in the Font Database plus the *raster font* files (see Figure 1-1).

Raster fonts are used by high-resolution monitors with the WYSIWYG feature (what-you-see-is-what-you-get) to display fonts on the screen.

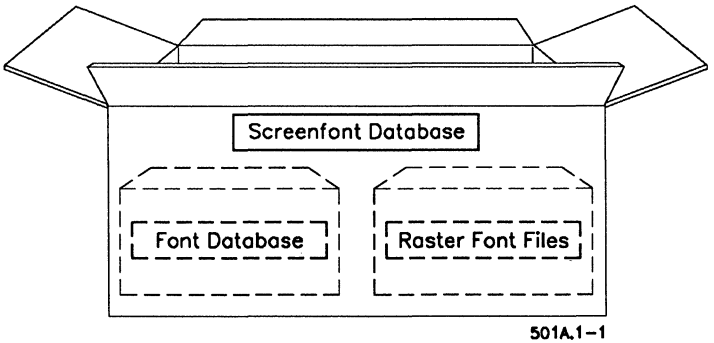


Figure 1–1. Screenfont Database

Editable Font Database

The *editable* font database is used as a storage facility for all your font data. As its name implies, the editable font database is also used to edit font data when you add new fonts or new printers. When you edit this database, you select the font data to be included in the runtime font database.

Runtime Font Database

The *runtime* font database is the active, working Font Database. Application software programs such as Document Designer and OFIS Designer use the font data in the runtime font database.

When you use the Font Tool to edit the Font Database, your goal is to make sure that the runtime font database contains all the font data necessary to print all the fonts that your printer can print. To improve performance, you also want to make sure the runtime font database contains only the data you need, and not, for example, font data for printers that you don't have.

The Font Services

To make use of the data in the runtime font database, you need the *Font Service*. (You install Font Service when you install GPS.) The Font Service is a system service that accesses the font data in the runtime font database for your application software and for the device drivers that control your printers. As mentioned earlier, this data is used to lay out text on the screen, to instruct the printer how to lay out text on the page, to make sure the right characters are printed, to inform the printer which font cartridge or print wheel to use, and which font to substitute if font substitution is required.

Now, you can also use the *Scaling Font Service* to access the font data in the runtime font database. The Scaling Font Service is distributed with the Generic Print System and consists of two elements: outlines that match the 35 fonts resident in LaserWriter II printers and a font database that supports the display of bit-maps to applications such as Document Designer and OFIS Designer. The bit-maps are available in any size, in any resolution, and for any orientation. To deliver this variety and flexibility, fonts are scaled on-the-fly.

Font substitution occurs when you request a font that your printer can't print. For example, certain printers can't print 13 point type, so if you request 13 point type, the Font Service will make a substitution, probably 12 point type.

If you've used Document Designer 2.0 or OFIS Designer 2.0, or higher you've seen font substitution in action. For example, if you specify 13 point Helvetica for text in your document, you may see this entry in the format status line at the top of the screen:

12 (13) point Helvetica

The font attribute in parentheses is the one you specified; the attribute outside the parentheses is the font substitution. In this case, only the point size is substituted, but it is possible that all the font characteristics might be substituted, including font family, point size, and type style. Figure 1-2 shows the format status line at the top of the screen.

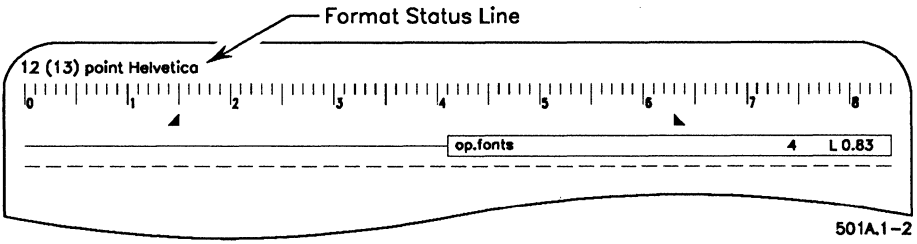


Figure 1–2. Document Designer Format Status Line

Font substitution gives you *printer independence*. For example, you can tell that this manual, with its sophisticated-looking typography, was printed on a laser printer. You could, however, print this manual on a daisy wheel printer or a dot matrix printer without changing fonts in the application. The pages wouldn't have all the varied fonts that you see in this manual; they would all be substituted with 12 point Courier and Courier bold or a similar font.

Applications and the Font Service

When an application program requires font data, it goes through the following steps, which are shown in Figure 1-3:

1. The application requests a font from the Font Service.
2. The Font Service searches for a font in the runtime font database that matches the request from the application.
3. The Font Service sends the application the data necessary to format the text.
4. The application sends the document to the device driver for printing. The device driver requests the font from the Font Service.
5. The Font Service sends the font data to the device driver.
6. The device driver uses the font data to control the printer so that the document is printed correctly.

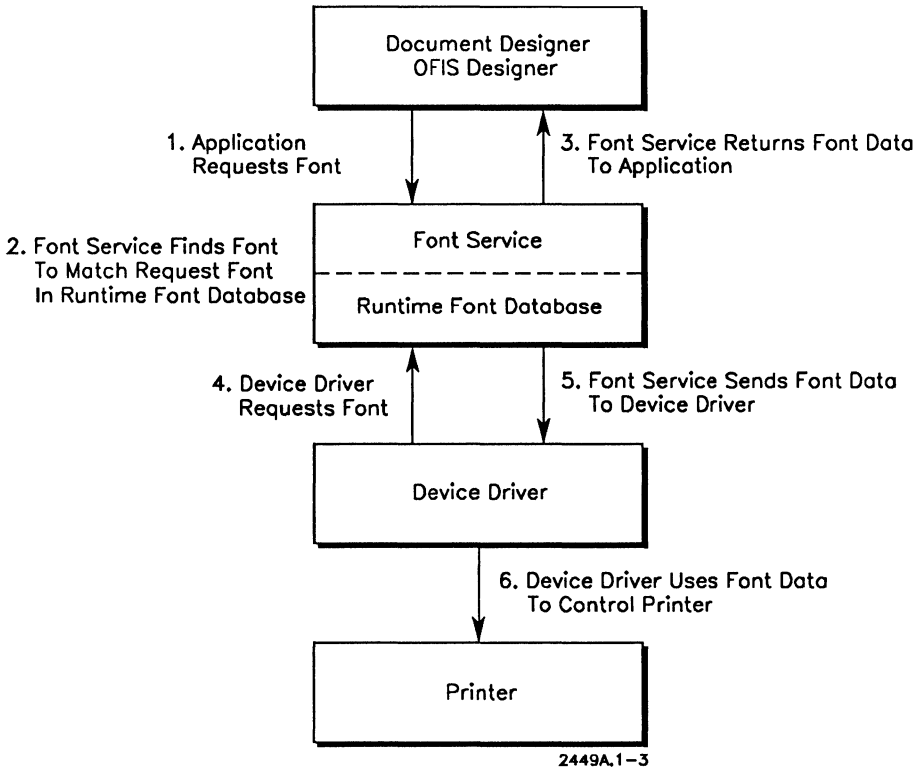


Figure 1–3. Document Designer, OFIS Designer, and the Font Service

Editing the Font Database

As you've seen, in order to print a variety of fonts, your software has to access the necessary font data. Your job is to make that font data accessible. This is accomplished in three basic steps:

1. Using the Font Tool to edit the editable font database.
2. Using the Font Tool to generate a new runtime font database, which incorporates the changes made to the editable font database.
3. Using the Executive to overwrite the old runtime font database with the new one.

For example, let's say you have a font cartridge that has a new font family called Letter Gothic. In order to use this font family, you need to add its font data to the runtime font database. To do so, you would follow the three steps below:

1. Use the Font Tool to add the Letter Gothic font data to the editable font database.
2. Use the Font Tool to generate a new runtime font database that includes Letter Gothic.
3. Use the Executive to overwrite your old runtime font database with the new one. You can now print Letter Gothic.

Figure 1-4 illustrates the above example. It shows the font families in the editable font database and the runtime font database. Notice that the new font family, Letter Gothic, has been added to both the editable font database and the runtime font database.

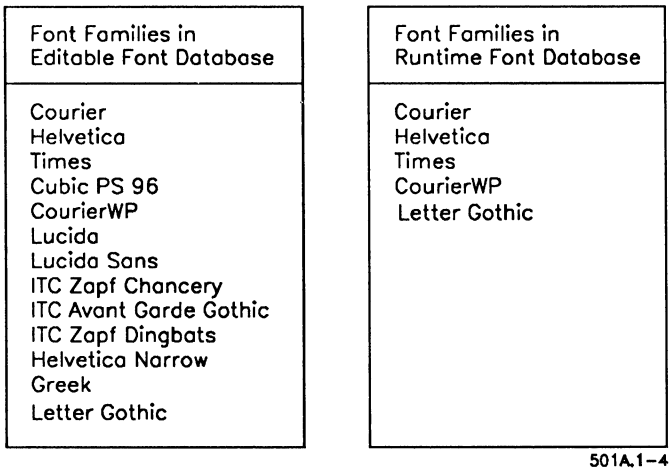


Figure 1–4. Adding a Font Family to the Runtime Font Database

Whereas you want to be able to take full advantage of your printer's capabilities, you want to do so as efficiently as possible. To accomplish this, you sometimes need to remove font data from the runtime font database. For example, let's say that the only font family you can use is Courier. In this case, all the other font families in the runtime font database are useless; they just use up disk space. You need to delete them from the runtime font database. To do so,

1. Use the Font Tool to "turn off" all font families except Courier in the editable font database.
2. Use the Font Tool to generate a new runtime font database that only includes Courier.
3. Use the Executive to overwrite your old runtime font database with the new one. You now have a smaller, faster runtime font database.

Figure 1-5 shows the font families in the editable font database and the new runtime font database after you've completed the above steps. Notice that when you edited the editable font database, you didn't delete any font families. All the other font families are still there, waiting for the day when you get a printer that can print them.

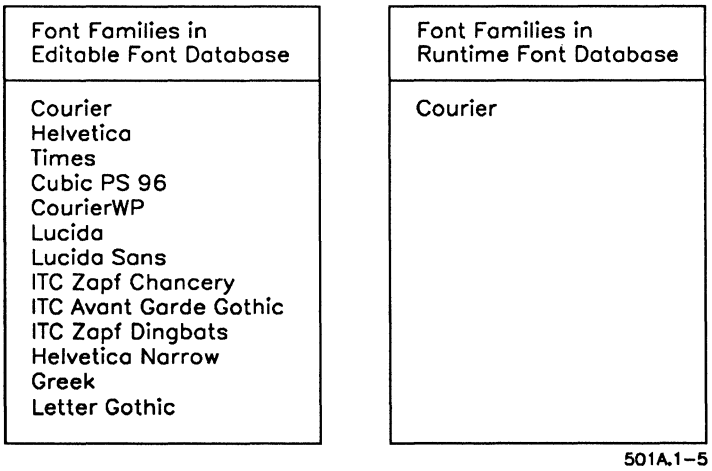


Figure 1–5. Removing a Font Family From the Runtime Font Database

How the Components of the Font Database Work Together

Remember that as you work with the Font Tool to make the best use of your application software and your printers' capabilities, you are often working at a level of detail that is remote from the day-to-day tasks of writing letters, creating business reports, and so forth. It is important, therefore, to understand how all the elements in the Font Database function together: the Font Database, your application software, the printers, and the Font Tool. This is described below and shown in Figure 1-6.

- You use the Font Tool to edit the editable font database, modify raster fonts, and generate the runtime font database.
- Application Programs, such as Document Designer and OFIS Designer, request font data from the Font Service.
- The Font Service retrieves the font data from the runtime font database and sends it to the application programs and the device drivers.

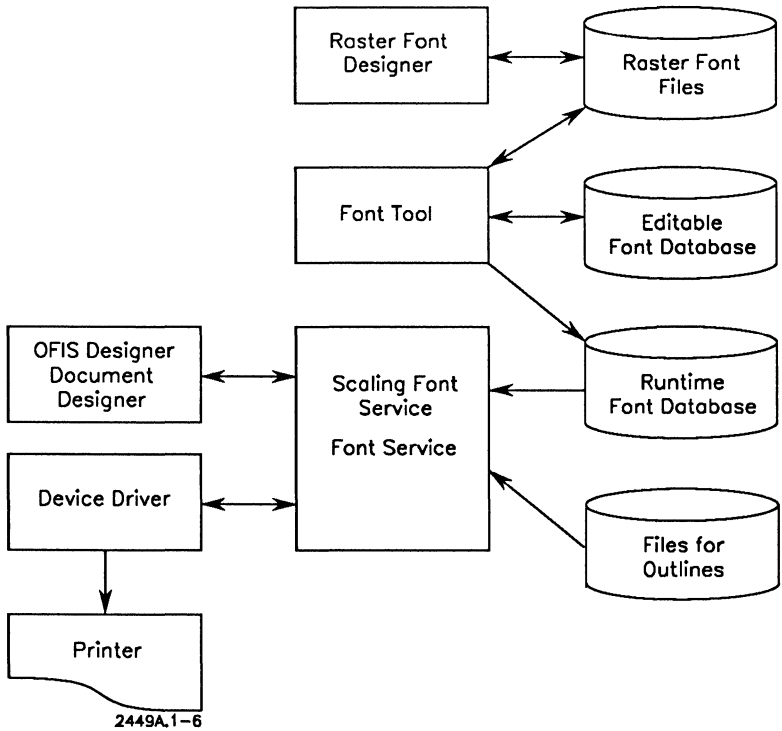


Figure 1–6. Overview of Font Database

Section 2
Inside the Font Tool

This section provides an in-depth look at the Font Tool. The procedures in the rest of this guide assume that you have read this section. If you are new to the Font Tool, it is strongly recommended that you read this section first.

If you want to enter the Font Tool to see what everything looks like as you read the material in this section, see "Entering the Font Tool," in Section 3. If you do so, you will need to use the **F5** function key to see some of the screens. To return to the main menu, use the **CANCEL** key to avoid saving any changes. Note that your editable font database may be customized, so the figures in this manual may not match exactly the displays on your screen.

Organization of the Editable Font Database

Font data is organized into *directories*. In the Font Tool, the word *directory* is roughly synonymous with *list*. For example, the Font Families directory is a list of font families. This use of the term *directory* is different from the way it is used in file specifications, for example, *[Sys]<FontDb>HelvBold14.font*.

A directory often provides access to a *detail form* or a table. For example, the Font Keys directory lists the names of the font keys, but each individual font key has its own detail form (see Figure 2-1).

Entries in the directories are called *records*. In Figure 2-1, the first record is for the daisy 10 point Courier font key. Each record may include several *fields*. In Figure 2-1, the first field in the record gives the name of the font, Courier.

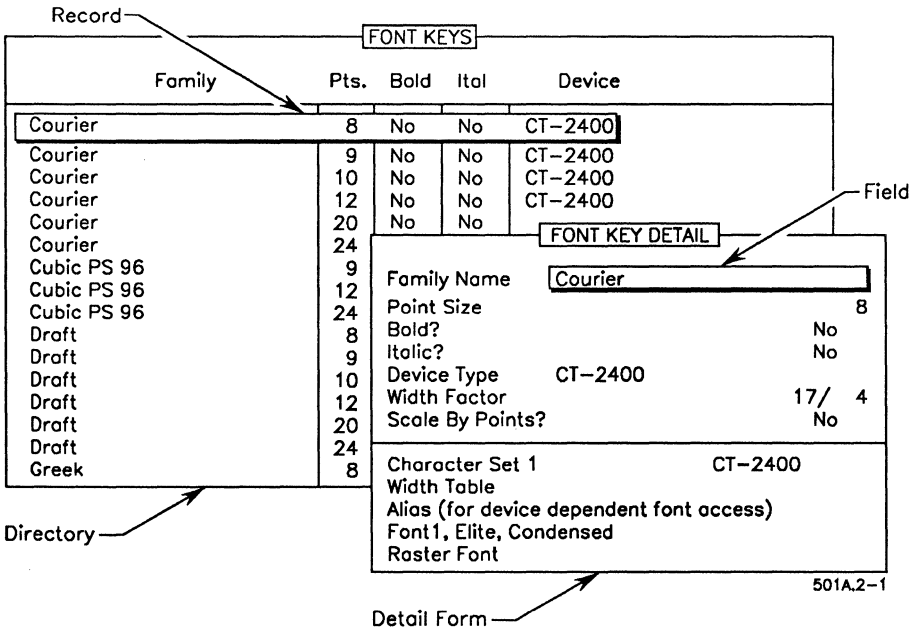


Figure 2–1. A Directory and a Detail Form

Variant Font Databases

Before you learn about the various tables and directories in the Font Tool, you should know about variant databases.

A variant database is a modified version of the editable font database. When you use a variant database, your modified records are kept separate from the unmodified records. Records that you modify are kept in a *variant database*; unmodified records are kept in the *original database*.

This means that if you modify a record in the variant database, the unmodified version of that record remains in the original database. The numerous advantages of this system are explained below, but first let's look at the concept visually.

Figure 2-2 shows the records in the original database, the variant database, and the runtime font database generated from the variant database. Notice that records A, B, and C are the only modified records, so they are the only records in the variant database. However, the runtime font database includes records from both the original database and the variant database.

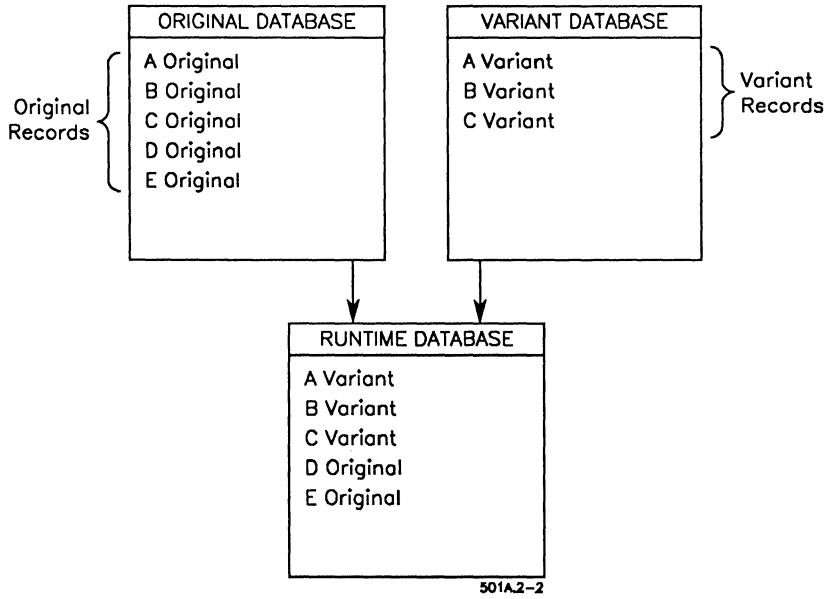


Figure 2–2. Variant and Original Databases

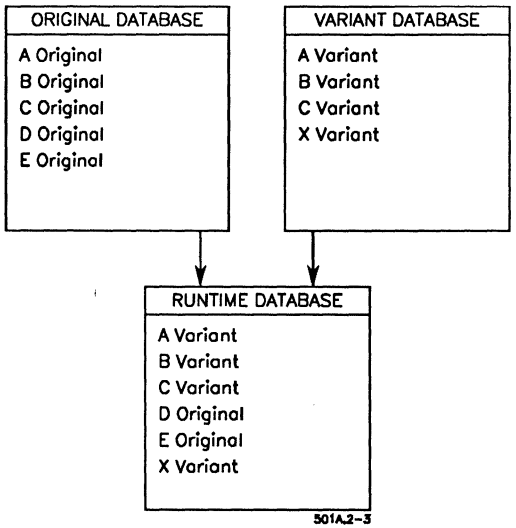


Figure 2–3. Adding to the Variant Database

In Figure 2-3, a new record has been added to the variant database. Note that the new record is included in the runtime database but is not included in the original database.

To use the font data from a variant database, you need to generate the runtime font database from that variant database (see Figure 2-3). In Figure 2-4, the runtime font database has been generated from the original database. Notice that none of the variant records are included.

There are additional variations on the information in the above figures, but first look at the advantages of using variant databases, outlined below.

- You can keep track of the changes you make to the editable font database.
- Changes won't be overwritten when you upgrade your GPS software.
- You can recover original data from the editable font database after you have modified it.
- You don't have redundant font data when you nationalize multiple font databases.

Examples of how these features are used are described in the next several sections.

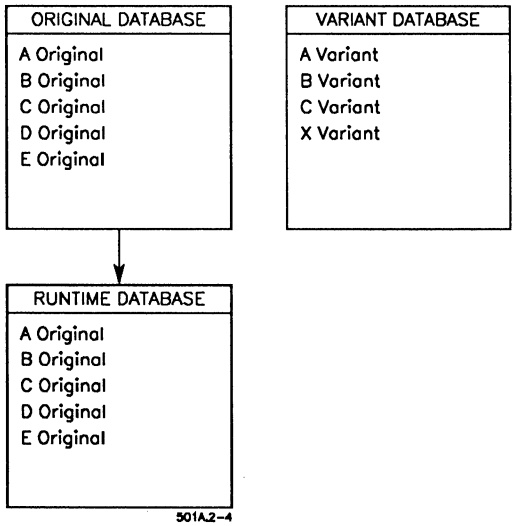


Figure 2–4. Runtime Database Generated From the Original Editable Database

Keeping Track of Changes

Max needs to modify his editable font database to include the font data for the HP LaserJet F font cartridge. (The standard version of the editable font database supports the B cartridge.) When Max makes his changes, he uses a variant database. As he makes the changes, each modified record is flagged with a plus sign (+), which identifies the records he has changed.

Preventing Changes From Being Overwritten

Now Max upgrades his Generic Print System (GPS) software. He overwrites the font data in the original database with the font data from the new software, but the modified data in the variant database remains untouched.

Figure 2-5 shows what happens when you upgrade your GPS software. Notice that when the runtime font database is generated, it includes the new records from the original database and the old records from the variant database.

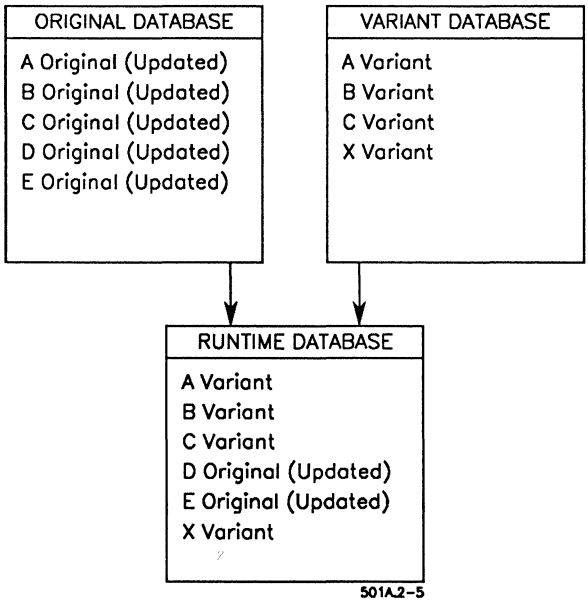


Figure 2–5. Upgrading the Editable Font Database

Recovering Original Data

Max loses his F cartridge. Fortunately, he still has a B cartridge. Since he made his earlier modifications to a variant database, his original database, containing the font data for the B cartridge, remains intact. This means that he can restore that font data quickly and easily.

Minimizing Redundant Font Data

Max finds himself in the business of distributing nationalized font databases. He has customers in Norway, Yugoslavia, and Angola. Since those countries all use different characters, they each need a separate font database.

However, there will be some font data that is common to all three countries, for example, the font data supporting certain printers. This shared font data will remain in the original database and will be used by all three variant databases. This not only avoids redundant data (thereby saving disk space), but it will also make it easier for Max to update the data shared by the variant databases. In addition, it is the only way to keep the font data particular to each nationalized character set separate from the other font data. Max will also be able to keep his own customized variant database, MaxFonts, separate from the others.

Figure 2-6 shows an example of an original database and two variant databases. Notice that the runtime font databases generated from the two variant databases include the shared unmodified data from the original database. Also notice that the two variant databases include records unique to each of them.

When to Use a Variant Database

You should always use a variant database whenever you make changes to the editable font database. There may be times when you need to edit the original database (as explained below), but those time will be rare.

You can use multiple variant databases as a quick and easy way to distribute specific font data. For example, you can create all the font data for a single printer in a variant database. You can then use one of the export commands to copy only the data pertaining to that printer to a set of port files, which you then use to distribute the data. If you find that you are creating font data for numerous types of printers, it is probably easiest to create a variant database for each one. This gives you the option of merging the font data for each printer into your own variant database as you need it.

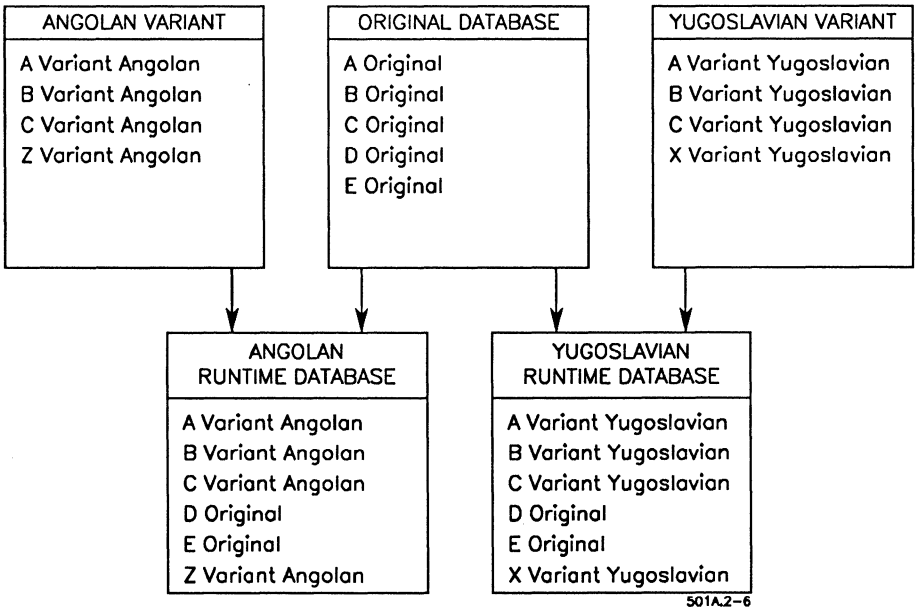


Figure 2–6. Sample of Two Variant Databases

Multiple variant databases are also used to nationalize font databases. As explained in the previous example (where Max created variant databases for Norway, Yugoslavia, and Angola), you would create a variant database for each nationalized font database.

It's important to remember that when you nationalize the Font Database, your work with the Font Tool is just part of the overall process. (If you aren't concerned with nationalization, skip ahead to the next section.) You also need to modify the system font so that the nationalized font can be used by all your software, for example, in command forms and menus. This system font will also be used in the tables and directories in the Font Tool. Normally, the Font Tool uses the system font *1024x768_80Sys.font* (for the high resolution monitor) or *T1Sys.font* (for all other monitors). If you nationalize the system font, you usually add an appropriate prefix to the system font, for example,

AngolaT1Sys.font. If you then name your variant database *Angola*, the Font Tool will reference *AngolaT1Sys.font* instead of *T1Sys.font*. This will make it much easier to work with translation tables.

When you nationalize numerous font databases, you sometimes need to modify the original database as well. For example, if you add a printer for all three variant databases, you would add the device type to the original database.

Note: *The procedures in this book assume that you are using a variant database. If you're familiar with the Font Tool, you may be accustomed to changing the names of records when you modify them, particularly in translation tables. When you use a variant database, however, it is best not to change the name of modified records because this makes it more difficult to restore the original records.*

The rest of this section explains the uses of each of the directories and commands in the editable font database. You may notice that the order in which these elements are discussed is different from the the order in which they are displayed on the main menu. This reordering is only in the interest of making the concepts easier to understand.

The Main Menu

The Font Tool main menu, as shown in Figure 2-7, provides access to the Font Tool directories. It is also used to start the Font Tool commands. (You need to scroll up to see some of the commands.) Note that the Export Variant command is displayed on the main menu only if you are using a variant database. Also, the name of the variant database is displayed at the top of the menu.

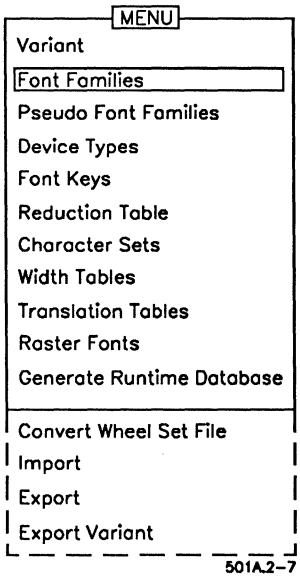


Figure 2–7. The Main Menu

Font Families

The Font Families directory lists the font families included in the editable font database, as shown in Figure 2-8.

FONT FAMILIES

Font Family	Gen?	Proportional?	Similar Font Family
Courier	Yes	No	
Helvetica	Yes	Yes	
Times	Yes	Yes	
Cubic PS 96	Yes	Yes	
CourierWP	Yes	No	
Greek	Yes	Yes	
Draft	Yes	No	Courier
Palatino	Yes	Yes	Times
ITC Avant Garde Gothic	Yes	Yes	Helvetica
ITC New Century Schoolbook	Yes	Yes	Times
Helvetica Narrow	Yes	Yes	Helvetica Narrow
ITC Bookman	Yes	Yes	New Century Schoolbook
ITC Zapf Chancery	Yes	Yes	Times
ITC Zapf Dingbats	Yes	Yes	
ITC Lubalin Graph	No	Yes	Courier
ITC Souvenir	No	Yes	ITC Bookman

501A.2-8

Figure 2–8. Font Families Directory

When to Use the Font Families Directory

If your printer can print a font family that isn't on the *Fonts* menu, you need to add that font family to the Font Families directory, with the following two exceptions:

- If the font family is already included in the Font Families directory under a different name. For example, Elite is just another name for Courier. You want to avoid adding redundant font families to the Font families directory because each time you add a font family, you need to add other kinds of font data: font keys, reduction rules, and so forth.
- If a font family is included in the *editable* font database but not in the *runtime* font database. If this is the case, you only need to modify the Font Families directory to include the font family in the runtime font database. To do this, you use the *Gen?* field in the Font Families directory. If you enter **yes** in this field, the font family will be included in the runtime font database.

Note: *You should also be aware that Document Designer and OFIS Designer display no more than 31 font family names. Therefore, set* Gen? *to* **no** *on font families that are not available on printers you can access.*

You can also modify the Font Families directory to remove unused font families from the runtime font database. Once again, you use the *Gen?* field; this time enter **no** in the field. By doing so, your Document Designer *Fonts* menu will be consistent with your printer capabilities, and you will also save disk space.

The Default Font Family

When you work with the Font Families directory, be aware that the first font family on the list is the default font family. When you create a new document, Document Designer uses the default font family to format your document until you specify otherwise. The default font family must be a monospaced font.

The only time the default font family is not the first font family in the list is when the first font family has the *Gen?* field set to *No*. In that case, the default font family is the first one in the list that has the *Gen?* field set to *Yes*.

Similar Font Families

Similar font families are used to automatically create reduction rules, the rules governing font substitution. (Reduction rules are discussed later in this section.) When you assign a similar font family, it will be the replacement font family if the requested font family is not available. For example, Figure 2-8 shows that the similar font family for ITC Souvenir is ITC Bookman. If ITC Bookman is not available, its similar font family is used, ITC New Century Schoolbook. Notice that if ITC New Century Schoolbook is not available, Times is used. However, since Times has no similar font family specified for it, the default font family is used, in this case, Courier.

You may have noticed that as you go from similar font family to similar font family, you move upward in the Font Families directory. This is not coincidental; in fact, when adding a font family to this directory, it is a good rule to choose a similar font family from an entry above the one you are adding. In this way, you avoid the possibility of a closed loop of font substitution, for example, Helvetica Narrow substituted for Helvetica, and Helvetica substituted for Helvetica Narrow, and so on.

Device Types

A font is usually defined by three characteristics: point size, type style, and font family. When you use the Font Tool, a fourth characteristic, the device type, is used to define fonts. For example, 12 point Times bold printed on an Imagen has a different appearance from 12 point Times bold printed on a PostScript printer. Therefore, when you read about fonts in this manual, they are often described like this: "Imagen 12 point Times bold," or "PostScript 12 point Times bold."

Device types are generally defined by the technology used to output fonts. For example, the *daisy* device type includes a number of printers that print with daisy wheels. In the case of laser printers, even though the technology is the same, the device type is determined by the device driver. For example, the PostScript device type includes several printers that use the PostScript device driver.

Device types are also defined by font capability. For example, the PostScript and LaserWriter+ device types use the same device driver, but they have different font capabilities. LaserWriter+ printers can print more fonts than the original PostScript printers, so two distinct device types have been created.

The device types included in the editable font database are listed in the Device Types directory, as shown in Figure 2-9.

DEVICE TYPES

Device Type	Gen?	Resolution Hor.	Resolution Ver.	Rasters?	Print?	Alias?	Scan?
Imagen	Yes	*	*	No	Yes	Yes	No
Daisy	Yes	*	*	No	Yes	Yes	No
EpFx286	Yes	*	*	No	Yes	Yes	No
PostScript	Yes	*	*	No	Yes	Yes	No
HPLaserJet	Yes	*	*	No	Yes	Yes	No
Simple	No	*	*	No	Yes	No	No
LaserWriter+	No	*	*	No	Yes	Yes	No
Video120	No	120	120	Yes	No	No	No

501A.2-9

Figure 2–9. Device Types Directory

When to Use the Device Types Directory

You need to add a device type if it is not included in the Device Types directory. This usually means that the new device type uses a different device driver or has different font capabilities than all the other device types. Remember that if you don't see the name of your printer in the Device Types directory, it may be included in one of the device types that support a variety of printers, for example, PostScript. To find out if your printer is included in a device type, see the section "Device Types in the Standard Editable Font Database," later in this section.

You need to *modify* existing device type records to add device types to or omit them from the runtime font database. For example, if you don't have an Imagen printer, deleting the Imagen font data from the runtime font database will make the database considerably smaller, which will save disk space. To do this, you use the *Gen?* field in the Device Types directory. If you enter **yes** in this field, the device type will be included in the runtime font database. To remove a device type from the runtime font database, enter **no** in the *Gen?* field.

The Default Device Type

When you work with the Device Types directory, be aware that the first device type in the list is the default, if the *Gen?* field for that device type is set to *Yes*. When you create a new document, Document Designer or OFIS Designer uses the default device type to format your document until you specify otherwise.

As with the default font family, if the first device type in the list has the *Gen?* field set to *No*, then the default device type is the next one in the list with the *Gen?* field set to *Yes*.

Downloadable Fonts

The standard editable font database contains the font data to support certain downloadable HP LaserJet fonts. (Downloadable fonts are font files that are not resident in the printer but instead are stored on your hard disk. When you want to use these fonts, they are sent, or *downloaded*, to your printer.) There is also a device type in the Device Types directory, HPDownLoad, that supports downloadable fonts. An example of adding downloadable fonts is included in Section 5.

The *Scan?* field in the Device Types directory is used for downloadable fonts. If you enter **yes** in the *Scan?* field, the device driver scans each page for downloadable fonts before sending the page to the printer. This is necessary for some printers, and unnecessary for others. Often, the only way to find out if the device driver needs to scan each page is to enter **no** in the *Scan?* field, and test the printer.

Device Types in the Standard Editable Font Database

The following list describes the device types included in the standard editable font database. Use this list to see if your printer is supported by the standard Font Database.

AP1324	Applies to the AP1324 in LQ2250 emulation.
AP9206	Applies to the AP9206.
AP9215	Applies to the AP9208, AP9215, AP9215-1, and AP9230.
APDMatrix	Applies to the AP1314, AP1327, AP1329, AP1351, AP1351-1, and AP1354 printers.
CT-2400	Applies to the CT-2425, CT-2450, and CT-2475.
Daisy	Applies to the AP1302, AP1303, AP1305, AP1307, Diablo 630 API-2, Diablo 630 HPR05, NEC 3515, and Qume Sprint 5, 9, and 11/55 Plus.
EPFX286	Applies to the Epson FX-286.
HPDownLoad	Supports HP LaserJet downloadable fonts sold by Hewlett Packard.
HP LaserJet	Applies to the HP LaserJet, HP LaserJet+, HP LaserJet 500, and HP LaserJet Series II.
HP PaintJet	Applies to the HP PaintJet printer.
HP Plotter	Applies to the HP7550A, HP7475A, and HP7470.
HP Scale	Used for scaled downloaded fonts.
Imagen	Applies to the Imagen IP2 and IP3.
LaserWriter+	Applies to LaserWriter+, LaserWriter II NTX, and other PostScript printers that print the larger set of font families (Palatino, ITC Avant Garde Gothic, and so forth).
LZ-05	Applies to the Convergent LZ-5 printer.

LZ-12	Applies to the Convergent LZ-12 printer.
PostScript	Applies to printers that use the PostScript device driver, specifically the AP9415, QMS, QMS PS-800, TI 2108, TI2115, and LaserWriter. This device type supports PostScript printers that print Courier, Times, and Helvetica.
Simple	Applies to Printronix MVP, Envision 420 and 430 Vectorprinters, and other draft printers.
Video120	VM-003 high-resolution monitor, used by screenfonts or the Scaling Font Service.

Translation Tables

Translation tables are used to make sure that when you enter a character at your keyboard you get the same character on the screen and on the printed page.

Every character that you enter at the keyboard is represented by a hexadecimal code, called the *character code*. You've probably already used hexadecimal codes in Document Designer and OFIS Designer. For example, when you want to type the bullet character (•), you may perform the following steps: Press **CODE-I**, enter **1B**, and press **GO**. 1Bh is the character code for the bullet.

Note: *Throughout this manual, to differentiate between decimal and hexadecimal notation for the character codes, an h is added to hexadecimal numbers, for example, 1Bh, as in the above paragraph. However, in the figures in the manual, the character codes are shown as they appear in the Font Tool, that is, without the suffix.*

When you list all of the character codes for a font, you have a *character code set*. Figure 2-10 shows part of the CTOS character code set, which is the one used by your application software. Your printers, however, have their own character code sets, and each one assigns character codes differently. For example, the character code for the bullet is 1Bh in the CTOS character code set, 0Dh in the Imagen character code set, and B7h in the PostScript character code set.

Character Code	Video Display Character	Character Code	Video Display Character	Character Code	Video Display Character	Character Code	Video Display Character
00h		20h		40h	@	60h	`
01h	↑	21h	!	41h	A	61h	a
02h	◣	22h	"	42h	B	62h	b
03h	¢	23h	#	43h	C	63h	c
04h	■	24h	$	44h	D	64h	d
05h	□	25h	%	45h	E	65h	e
06h	½	26h	&	46h	F	66h	f
07h	🔔	27h	'	47h	G	67h	g
08h	⇤	28h	(	48h	H	68h	h
09h	⇥	29h	)	49h	I	69h	i
0Ah	♪	2Ah	*	4Ah	J	6Ah	j
0Bh	↓	2Bh	+	4Bh	K	6Bh	k
0Ch	↖	2Ch	,	4Ch	L	6Ch	l
0Dh	◢	2Dh	–	4Dh	M	6Dh	m
0Eh	←	2Eh	.	4Eh	N	6Eh	n
0Fh	‡	2Fh	/	4Fh	O	6Fh	o
10h	¼	30h	0	50h	P	70h	p
11h	↕	31h	1	51h	Q	71h	q
12h	→	32h	2	52h	R	72h	r
13h	⌙	33h	3	53h	S	73h	s
14h		34h	4	54h	T	74h	t
15h	÷	35h	5	55h	U	75h	u
16h	ı	36h	6	56h	V	76h	v
17h	$	37h	7	57h	W	77h	w
18h	≠	38h	8	58h	X	78h	x
19h	≃	39h	9	59h	Y	79h	y
1Ah	¶	3Ah	:	5Ah	Z	7Ah	z
1Bh	●	3Bh	;	5Bh	[	7Bh	{
1Ch	¬	3Ch	<	5Ch	\	7Ch	|
1Dh	≤	3Dh	=	5Dh	]	7Dh	}
1Eh	±	3Eh	>	5Eh	^	7Eh	~
1Fh	≥	3Fh	?	5Fh	_	7Fh	■

501A.A-1.1

Figure 2–10. CTOS Character Code Set (00h–7fh)

Translation tables act as interpreters between character code sets. In Document Designer, when you want a bullet, you enter **1B** at the keyboard. The Font Service then uses the translation tables to translate 1Bh into the character code used by the specified printer. If it's a PostScript printer, the Font Service translates 1Bh to B7h. This process is called *character code translation.*

Unlike font families and device types, translation tables function at a level that is transparent to the user. Therefore, each time a user wants to insert a bullet in a document, he or she doesn't have to remember the character code for a bullet for every single printer. The user just enters **1B**, knowing that the Font Service will use the translation tables to get the correct character code to the printer.

Note: *You may want to look at the character code set for your printer to compare it with the CTOS character code set. (The printer's character code set should be included in your printer documentation.) It may, however, show the character codes as decimal numbers instead of hexadecimal codes. If this is the case, you need to convert the decimal numbers into hexadecimal numbers when working with the Font Tool. Appendix C provides a decimal/hexadecimal conversion table.*

When to Add or Modify Translation Tables

You need to add a translation table whenever you add a new device type. (The only exception is when the new device type is simply a variation of an existing device type; for example, the PostScript and LaserWriter+ device types both use the same translation table.) When you add a translation table for a new device, you are adding the character code set for the new device to the Font Database so the Font Service can make the character translations necessary for the new device.

You also need to add translation tables when you add symbol fonts that aren't included in the standard editable font database. For example, if you add the HP LaserJet Linedraw font, you need to add a translation table to access the characters in that font.

You need to modify translation tables when you want to print characters that are resident in your printer but are not included in the standard Font Database. These characters vary from printer to printer, but they are usually accented characters, such as E umlaut, or symbols, such as hearts and stars.

The Translation Tables Directory

The Translation Tables directory, as shown in Figure 2-11, provides access to individual translation tables.

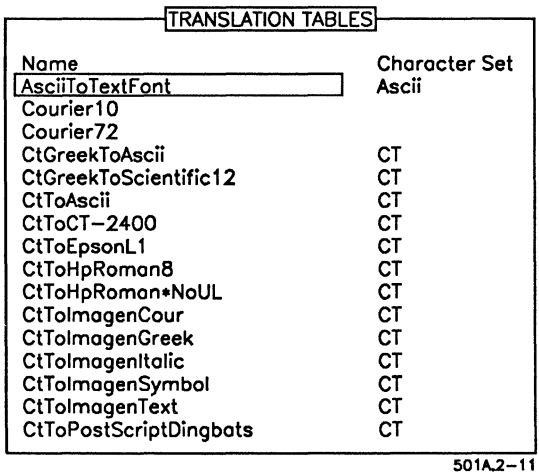


Figure 2–11. Translation Tables Directory

Most translation tables are *first level* translation tables. First level translation tables translate the CTOS character code set into the character code set of a particular device type. For example, *CTtoHpRoman8* translates the CTOS character code set into the HP LaserJet character code set.

Second level translation tables translate the printer's character code set into special printer codes. Normally, character codes are just shorthand ways of representing 8-bit binary numbers. (For example, the character code 4Dh is the hexadecimal equivalent of the binary number 01001101.) Some printers, however, require 7-bit or 16-bit binary numbers. Second level translation tables translate 8-bit codes into 7-bit or 16-bit codes. For example, *ImagenTextToGascii* translates Imagen character codes into 16-bit codes that the Imagen printer needs in order to print.

Usually a translation table supports a number of fonts that share a device type. For example, the *CtToPostScriptN* translation table is used by all PostScript text fonts. There is also a translation table for PostScript symbol fonts. Translation tables for symbol fonts are necessary because symbol fonts have different characters than text fonts and therefore different character code sets. For information on how each translation table in the standard editable font database is used, see Appendix D, "Translation Tables and Character Sets."

Sample First Level Translation Table

Figure 2-12 shows the first level translation table *CtToPostScriptN*. As indicated by the name, this table is used to translate the CTOS character code set to the PostScript text character code set.

The lower part of the table contains the translations of the characters. Since there can be as many as 256 characters, there isn't room to see them all on one screen; scroll up to see the rest of the characters.

CHARACTER TRANSLATION TABLE

Name: CtToPostScriptN
Character Set: CT

Comment:
Translates to Character Set: PostScriptN

18	≠		≃		¶	B6	•	B7
1C	¬		⩽		±		⩾	
20		20	!	21	"	22	#	23
24	$	24	%	25	&	26	'	27
28	(	28	)	29	*	2A	+	2B
2C	,	2C	-	2D	.	2E	/	2F
30	0	30	1	31	2	32	3	33
34	4	34	5	35	6	36	7	37

501A.2-12

Figure 2–12. Sample First Level Translation Table

Let's take a closer look at the elements of a translation table and how a character translation takes place. In Figure 2-12, find the bullet character at the end of the first row in the lower part of the table. The number to the right of it, B7, is the PostScript character code for the bullet. We already know that the CTOS character code for the bullet is 1Bh. Therefore, the character code translation for the Imagen bullet is from 1Bh to B7h. When you enter **1B** at the keyboard, the Font Service translates the character codes and sends B7h to the printer.

The codes that run down the left side of the table are the CTOS character codes for the first character in each row. For example, the CTOS character code for the not equal symbol (≠) is 18h. The CTOS character codes for the rest of the characters can be deduced by counting to the right. For example, the CTOS character code for the approximation symbol (≈) is 19h; the CTOS character code for the paragraph symbol (¶) is 1Ah; and the CTOS character code for the bullet is 1Bh.

Notice that some of the characters don't have a code to the right, for example, the plus or minus symbol (±). In this case, the plus or minus symbol is not mapped by this translation table, which means that it is not included in the PostScript text character code set.

Notice also that starting at the CTOS character code 20h, the PostScript and the CTOS character code sets are identical. For example, the exclamation point (!) has the same character code in both character code sets, 21h. In fact, most character code sets use the same character codes in the range 20h through 7Eh. This is called the ASCII range. ASCII standardizes the character codes of the most common alphanumeric and punctuation characters. Since most character sets use ASCII code, most of the work you do with translation tables will be outside the 20h through 7Eh range.

Sample Second Level Translation Tables

Figure 2-13 shows the second level translation table that translates certain Imagen character codes into 16-bit printer readable codes. Since this translation table translates *from* the Imagen character code set, the characters in the table are laid out in the order of the Imagen character code set, not the CTOS character code set. For example, the Imagen character code for the bullet is 0Dh, so in this translation table the bullet is in the 0Dh location. In first level translation tables, the bullet is always in the 1Bh location.

CHARACTER TRANSLATION TABLE

Name: ImagenTextToGascii Comment: maps to native character codes
Character Set: CT Translates to Character Set:

0C	% 02EF41	• 02EF66	ƒ 02EFA2	ff 02F021
10	ffi 02F022	ffl 02F023	fi 02F024	fl 02F025
14	Å 02F128	Ç 02f120	å 02F1A8	ç 02F1AD
18	– 02213E			
1C				
20		! 21	22	# 23
24	24	% 25	& 26	27
28	(28	) 29	* 2A	+ 2B

501A.2-13

Figure 2–13. Sample Second Level Translation Table

You only need to create a new second level translation table if you add a new device type that uses other than 8-bit character codes. For more information, see your printer documentation.

Translation Table Escapes

There are some hexadecimal codes that are not used to represent characters, but are instead used as escapes. *Escapes* are special codes used in translation tables to alter the way characters print or to work around the normal rules of translation tables. Appendix E lists the escapes and what they are used for, but let's look at one example right now.

The escape 10h is used to alter the print widths of characters, as follows: These are periods (.....), and these are leader dots (. . . .). As you can see, leader dots are spaced farther apart than periods. The spacing of leader dots is accomplished with the 10 escape.

The character code for the leader dot is 2D105F. 2D is the character code for the period (.), 10 is the escape that means "use the width of," and 5F is the character code for the underline character (_). When the codes are combined, they are translated as, "Print the period using the width of the underline."

Escapes can be found in first level and second level translation tables. Escapes in first level translation tables are used by the Font Tool, usually to manipulate the way characters are printed, as shown in the example above. Escapes in second level translation tables are used by device drivers to give certain instructions to the printer.

Character Sets

Character sets provide a quick way for the Font Service to access related translation tables. For example, Imagen fonts use both a first level and a second level translation table. To make it easier to access these two related translation tables, they are grouped together into one character set.

It is easy to confuse the terms *character set* and *character code set*. Character code set refers to a set of hexadecimal *codes*; character set refers to a *set* of translation tables.

When to Add or Modify Character Sets

Since character sets are comprised solely of translation tables, you only need to add or modify them if you add, remove, or change the name of a translation table. Changing the name of a character set is not recommended because if you do, you have to change the name in all the font keys that reference that character set.

The Character Sets Directory

The Character Sets directory, as shown on the left in Figure 2-14, provides access to the character set detail forms. For information on how each character set in the standard editable font database is used, see Appendix D.

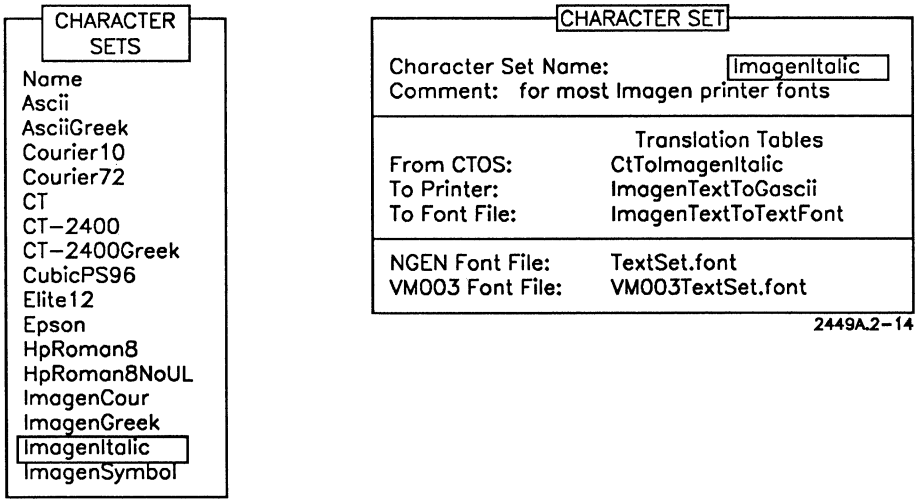


Figure 2–14. Character Sets Directory and Detail Form

Sample Character Set Detail Form

Figure 2-14 shows the detail form for the character set *ImagenItalic*, which is the character set used by Imagen italic text fonts. This example includes three translation tables, as described below.

From CTOS: The first level translation table that translates from the CTOS character code set to the device character code set. In this example, *CtToImagenItalic* translates from the CTOS character code set to the character code set used by Imagen italic text fonts.

To Printer: The second level translation table. In this case, *ImagenTextToGascii* translates the Imagen character codes into printer-readable 16-bit "Gascii" code.

To Font File: This is an optional field that allows you to use the font files listed at the bottom of the character set detail form, *TextSet.font* (for character mapped monitors) and *VM003TextSet.font* (for high-resolution monitors). These font files contain characters that are included in printer character code sets, but not in the CTOS character code set, for example ligatures (*ff*, *fi*, and so forth). These font files are used when displaying second level translation tables and width tables. Since they show the characters that the printer can print, it's easier to work with those tables. These font files have nothing to do with printing.

The *To Font Files* field is necessary because more than one device type uses these font files. This example lists the *ImagenTextToTextFont* translation table, but the *PostScriptN* character set lists the *PostScriptToTextFont* translation table in this field. In any case, the translation table listed in the *To Font Files* field translates the printer's character code set to the character code set used by the font file.

Width Tables

Width tables contain the widths of all characters in a font. They are used to make sure that text is laid out correctly on the page. Width tables are required for all proportional fonts; they are not used for monospaced fonts.

You can't use width tables to change the actual widths of characters; these widths are defined by the printer. The device driver uses character widths to instruct the printer where to print each character. An erroneous width table will cause the letters to be printed too close together or too far apart.

When to Add Width Tables

You need to add width tables whenever you add proportional fonts to the editable font database. For example, if you add an HP LaserJet font cartridge that includes 12 point Helvetica bold, you need to add a width table to be able to print that font.

The Width Tables Directory

The Width Tables directory, as shown in Figure 2-15, lists all the available width tables. The character set next to the name shows which character code set the width table is based on. If the width table is for an Imagen font, the table will be based on an Imagen character code set.

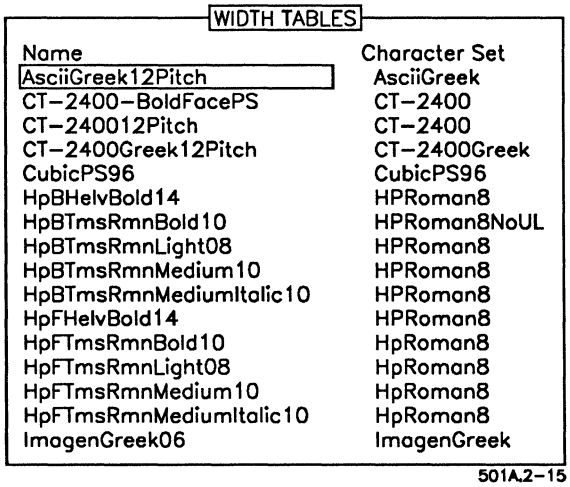


Figure 2–15. Width Tables

Sample Width Table

Figure 2-16 shows the width table *HpFHelvBold14*. As the name indicates, this table is for the 14 point Helv bold font from the HPLaserJet F cartridge, which is similar to Helvetica.

CHARACTER WIDTH TABLE

Name: HpFHelvBold14 Comment:

Unit of Measure: 300th of a inch Character Set: HPRoman8

58	X	42	Y	41	Z	39	[	22	\	31	]	20	^	33	_	47
60	`	18	a	31	b	34	c	29	d	33	e	29	f	20	0	33
68	h	34	i	15	j	20	k	32	l	15	m	52	n	34	o	33
70	p	34	q	34	r	22	s	29	t	24	u	34	v	32	w	48

501A.2-16

Figure 2–16. Sample Width Table

Each character in the font is shown in the lower part of the sample width table in Figure 2-16. The number to the right of each character is its width. For example, the number to the right of *X* is 42. The *Unit of Measure* field in the top part of the table shows the unit of measure for this table, which is 300ths of an inch. This means that *X* is 42/300ths of an inch wide. The unit of measure can be in either points or inches.

The numbers that run down the left side of the table indicate the character code of the first character in the row. For example, the character code for the *X* is 58h. To find the location of the rest of the characters in the row, count over to the right. For example, the character code for *Y* is 59h, the character code for *Z* is at 5Ah, and so on.

The characters in width tables are laid out in the ordering of the printer's character code set, not the ordering of the CTOS character code set. For example, if the width table is for a PostScript font, the characters will follow the ordering of the PostScript character code set. This makes it easy to transfer the widths of the characters from your printer documentation into the width table.

Font Keys

Font keys are the basic unit of data in the Font Database. All the data that your application software and your device drivers need is found in or pointed to by the font keys, including the attributes of the font (size, type style, and so forth), as well as the widths of the characters, the proper codes needed to print each character, and any information needed by the printer, for example, the names of font cartridges or print wheels.

Font keys are the focal point for all the font data in the Font Database. Figure 2-17 shows the data that is included in a font key from all the other directories in the Font Database. When an application requests a particular font, the Font Service first searches for font data in the font key of the requested font. The font key in turn refers to the correct width table, translation tables, and raster font.

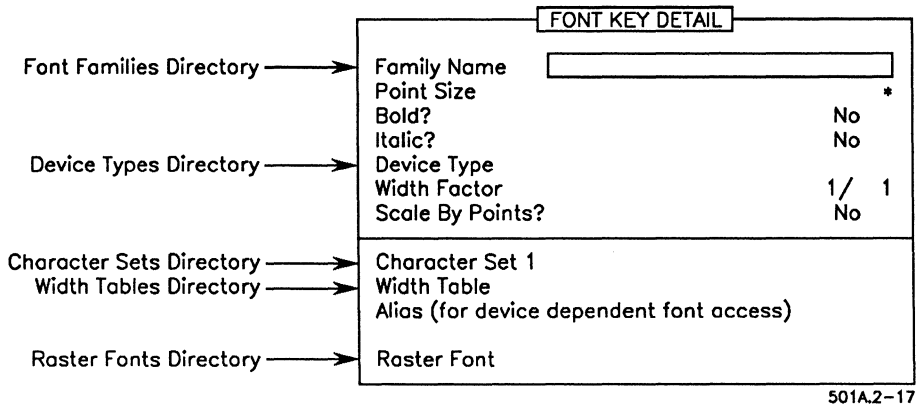


Figure 2–17. Font Key Data

When to Add or Modify Font Keys

You can't print a font without a font key. Therefore, you must have a font key for every font your printer is capable of printing. Whenever you add new fonts, you need to add new font keys.

When you add a font family, you need to add a font key for every printable combination of point size, type style, and device type. For example, if you add a font family that comes in only one point size, one type style, and can only print on one device type, you only need one font key.

The only exception is if you are adding a font family to a PostScript or LaserWriter+ device type (or for a font scaled by the Scaling Font Service). Fonts for these device types are scaled by the printer. Therefore, when you specify 12 point Times on a PostScript printer, it is sent to the device driver as 1 point Times and enlarged by the printer to 12 point Times. So, when you add a new PostScript or LaserWriter+ font family, you only need to create font keys for 1 point type, with variations of bold, italic, bold italic, and roman.

The Font Keys Directory and Detail Form

The Font Keys directory, shown in Figure 2-18, provides access to the font key detail forms.

A sample Font Key detail form is shown in Figure 2-19. The first five fields in the detail form, *Family Name*, *Point Size*, *Bold?*, *Italic?*, and *Device Type*, define the font attributes and are self explanatory. The other fields deserve some explanation.

FONT KEYS

Family	Pts.	Bold	Ital	Device
Courier	8	No	No	CT-2400
Courier	9	No	No	CT-2400
Courier	10	No	No	CT-2400
Courier	12	No	No	CT-2400
Courier	20	No	No	CT-2400
Courier	24	No	No	CT-2400
CubicPS96	9	No	No	CT-2400
CubicPS96	12	No	No	CT-2400
CubicPS96	24	No	No	CT-2400
Draft	8	No	No	CT-2400
Draft	9	No	No	CT-2400
Draft	10	No	No	CT-2400
Draft	12	No	No	CT-2400
Draft	20	No	No	CT-2400
Draft	24	No	No	CT-2400
Greek	8	No	No	CT-2400

501A.2-18

Figure 2–18. Font Keys Directory

Width Factor — Gives the width, in points, of each character. For monospaced fonts, the width of the character is represented by the fraction *72/Pitch*. For example, 12 pitch type is 72/12; 16 pitch is 72/16; and so forth. The width factor is used in the same way as width tables, to lay out text on the screen and to make sure each letter prints in the proper location on the page.

Proportional fonts use width tables to define character width, so the width factor is usually set to 1/1. This leaves the values specified by the width table unchanged. (A width factor of 2/1 would multiply each width in the width table by two.)

Scale by Points? — Applies only to device types that scale fonts from a single size. PostScript, LaserWriter+, Video120, and HpScale device types are the only device types in the standard Font Database that scale fonts by points.

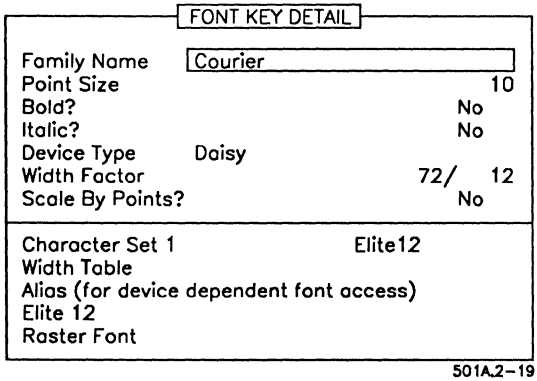


Figure 2–19. Sample Font Key

Character Set

Gives the name of the character set that contains all the necessary translation tables. There can be up to four character sets in a font key. Character set 1 is the *primary character set*, character set 2 is the *first alternate character set*, and so forth. (Scroll up to see all four character sets.) Alternate character sets are explained in the next section.

Width Table

Displays the width table for proportional fonts.

Alias

The information entered depends on the font key's device type. Each device type uses this field differently.

For example, if you have a document that is formatted using two different font cartridges, you will need to change cartridges while the document is printing. The name of the font cartridge that you enter in the *Alias* field will be displayed in a prompt on your screen that tells you to change font cartridges.

For more information on how each device type uses the *Alias* field, see Appendix B.

Raster Font

Applies only to font keys for the video120 device type (the font keys for raster fonts). This field displays the name of the raster font file.

Alternate Character Sets

By using alternate character sets, you can enter text in different fonts without having to use the Document Designer *Fonts* menu to change fonts. For example, this text is in the printer's 10 point Times text font, but this Up Arrow symbol (↑) is included in the printer's symbol font. Yet, when you enter the Up Arrow symbol, you don't have to change the font in Document Designer; 10 point Times works for both text and symbols. This sounds a little like magic, but actually it's a fairly simple process, as described below.

Figure 2-20 shows a three dimensional view of how alternate character sets work. The translation table at the top is the translation table for the *primary* character set, ImagenText. This is the character set used by 10 point Times text font. The translation table on the bottom is the translation table for the *first alternate* character set, ImagenSymbol. This is the character set used by the Up Arrow symbol.

The arrows in Figure 2-20 represent requests for characters. When you want an Up Arrow symbol in your Document Designer text, you press **CODE-I**, and then type **01** at the keyboard. The Font Service searches in the translation table for a character code at the 01h position. In this case, there isn't one, so the request is allowed to fall through the primary translation table to the first alternate translation table. Here there *is* a character translation for the Up Arrow symbol so the Font Service instructs the device driver to switch fonts and sends that character code to the device driver, and the Up Arrow symbol is printed.

Figure 2-20 also shows what happens when a request for a character is met by the primary character set. In such cases, the request is not allowed to fall through to the first alternate character set.

On the other hand, if no translation is made in the primary or first alternate character sets, the request falls through to the second alternate character set, if there is one, and if necessary, to the third alternate character set. (You can have as many as three alternate character sets.) If no character code for a character is found in any alternate character set, no character will be printed.

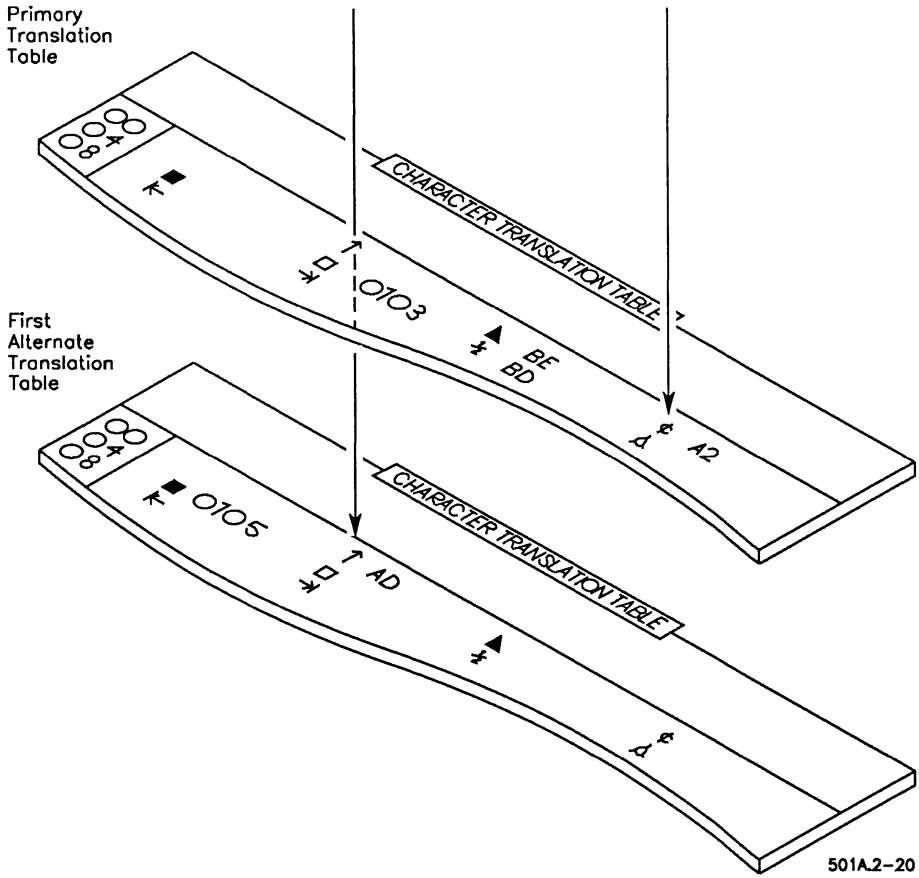


Figure 2–20. Primary and Alternate Character Sets: A Three Dimensional View

As mentioned earlier, the advantage of this system is that it permits you to easily access characters in different fonts. Figure 2-21 shows a font key with one alternate character set. Notice that the two character sets have different width tables and aliases, indicating that they are indeed two different fonts.

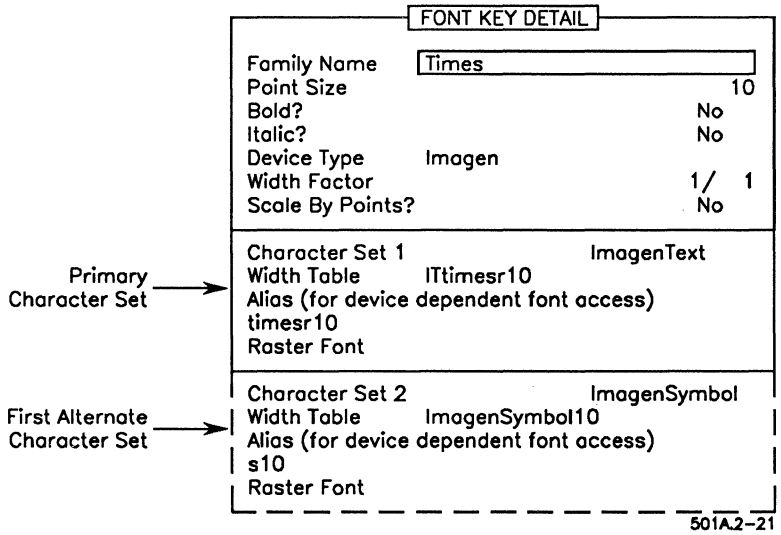


Figure 2–21. Primary and First Alternate Character Sets

Raster Fonts

This section discusses raster fonts, the fonts that you see on a high resolution monitor when you use the Document Designer WYSIWYG feature. If you do not use raster fonts, skip ahead to the next section.

Raster fonts include the same kind of font data as printer fonts. There are font keys for raster fonts, as well as reduction rules, translation tables, and so forth. The main difference is that raster fonts have a video monitor as the output device.

When to Modify Raster Fonts

The Font Tool permits you to tune the raster fonts distributed with the Screen Font Database to adjust how they are laid out on the screen. Adjusting the raster fonts does not affect how the corresponding text font is printed. You only tune raster fonts so that they appear the same on your screen as they do on the printed page. In the standard Font Database, raster fonts are tuned to Imagen printer fonts. If you use another kind of printer, you may want to tune raster fonts to match that device type.

Note: *Since you cannot change fonts scaled on-the-fly, do not attempt to tune raster fonts if you use the Scaling Font Service.*

When you tune raster fonts, you have two goals:

- To get a close *overall* match between the width of the printer characters and the width of the raster font characters.
- To make the raster fonts as aesthetically pleasing as possible, avoiding pairs of characters that are too close together or too far apart.

You cannot use the Font Tool to create new raster fonts or modify the shapes of the raster font characters. For that, you need the Raster Font Designer. In theory, you could create a new raster font and use the Font Tool to add the data to the Font Database. For example, you could create a raster font for ITC Zapf Chancery, but it would be a very large undertaking because each character needs to be drawn, literally bit by bit, and there are approximately 3,000 characters per font family. It is more likely that you would create raster images for individual characters. For example, if you use a lot of stars, you might want to create a raster character of a star. See the *Raster Font and Icon Designer Manual* for further information.

The Raster Fonts Directory and Detail Form

The Raster Fonts directory, shown in Figure 2-22, lists the available raster fonts. Text raster fonts use the *Video* character set; symbol raster fonts use the *VideoSymbols* character set. These character sets translate the *CT* character code set to the raster font character code set. If you used the Scaling Font Service, the Raster Fonts directory contains only those fonts that you prescaled (for faster access) using Export Raster Fonts.

The only unusual raster font is ConvergentR12.font, which uses the VideoCT character set. This is the raster font for the system itself and is used to display the line drawing characters when you select a 12 point font in Document Designer or OFIS Designer. (This applies to the Screen Font Database and not to the Scaling Font Service.)

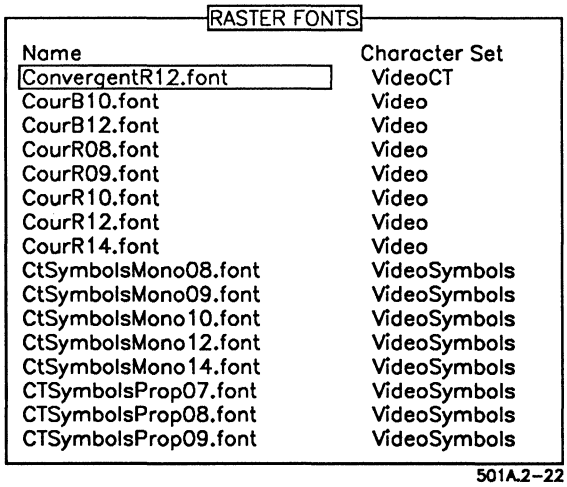
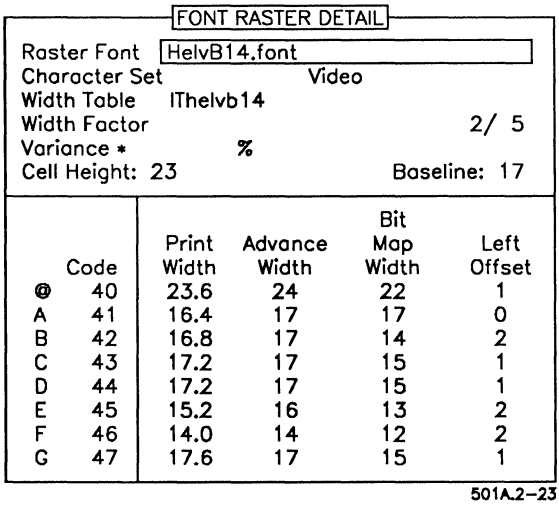


Figure 2–22. Raster Fonts Directory

The *Font Raster Detail* form, shown in Figure 2-23, is the form you work with when you tune raster fonts. See the section on tuning raster fonts in Section 3 for descriptions of each of the fields in this form.


FONT RASTER DETAIL

Raster Font HelvB14.font
Character Set Video
Width Table IThelvb14
Width Factor 2/ 5
Variance * %
Cell Height: 23 Baseline: 17

	Code	Print Width	Advance Width	Bit Map Width	Left Offset
@	40	23.6	24	22	1
A	41	16.4	17	17	0
B	42	16.8	17	14	2
C	43	17.2	17	15	1
D	44	17.2	17	15	1
E	45	15.2	16	13	2
F	46	14.0	14	12	2
G	47	17.6	17	15	1

501A.2-23

Figure 2–23. Raster Font Detail Form

Psuedo Font Families

Word Processor and Document Designer 1.0 use wheel sets to organize and store font data. If you have files created in WP or Document Designer 1.0, the font data in the wheel sets needs to be transferred into the Font Database so that you can format and print your old documents with Document Designer 2.0 or higher. The transferring of the font data from wheel sets to the Font Database is called *wheel set conversion*. During this process, pseudo font families are created. Pseudo font families map wheel names to font keys. In other words, they provide a link between two ways of handling font data: wheel set files, and font keys. This is explained in greater detail below.

When to Add Psuedo Font Families

You only need to add pseudo font families if you convert wheel sets. If you have a new print wheel or a thimble for which you don't have a wheel set file, you don't need to create a wheel set file and then convert it. You can just enter the font data (font family, translation tables, width tables, and so forth) directly into the Font Database the same way you would if you were adding a new font cartridge or printer.

The Psuedo Font Families Directory and Detail Form

The Pseudo Font Families directory, as shown in Figure 2-24, provides access to the pseudo font families detail forms.

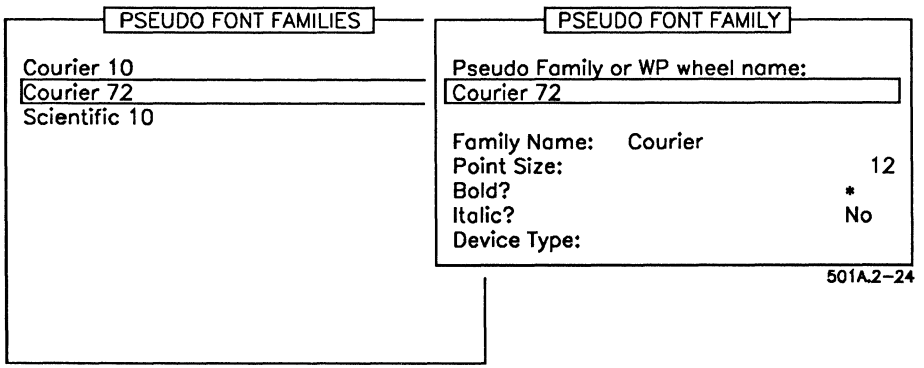


Figure 2–24. Sample Psuedo Font Families Directory and Detail Form

The pseudo font family detail form, also shown in Figure 2-24, is used to map the pseudo font family to a font key. Let's take a closer look at this process. First, remember that it is best to avoid redundant font families in the Font Families directory. For example, Courier 72 is just a variation on Courier, so you wouldn't have an entry for Courier 72 in the Font Families directory.

Now, let's say that you open a Word Processor document in Document Designer 2.2 or an OFIS Writer document in OFIS Designer 2.2. The document was originally formatted using the Courier 72 wheel set, so the requested font will be Courier 72. Since there is no Courier 72 font family, the Font Service searches in the pseudo font families. There is a Courier 72 pseudo font family, so the Font Service will use the data in the pseudo font family detail form to map the font request to a Courier font key.

Figure 2-24 shows the pseudo font family detail form for Courier 72. When the font family Courier72 is requested, the Font Service uses this detail form to map the request to a 12 point Courier font key. Notice that the *Bold?* field contains an asterisk. This is a wild card, which is filled in by the requested font if it is bold. The *Device Type*: field is left blank, so the device type can be filled in by the requested font as well.

Reduction Rules

Applications such as Document Designer allow you to choose among literally hundreds of fonts. What you can actually print, however, is ultimately controlled by the capabilities of your printer. Reduction rules allow you to choose any font you want with the knowledge that if your printer can't print it, it will print the closest font possible. As discussed in Section 1, this process is called font substitution.

Reduction rules follow this basic format: "When Font X is specified but is not available, substitute Font Y." For example, one of the simplest rules is, "When the font family ITC Zapf Chancery is specified on an Imagen printer, substitute Times." This rule indicates that ITC Zapf Chancery is not available on the Imagen printer.

Reduction rules are created automatically. If there is no font key for a font, a reduction rule is created to provide a replacement font. For example, if there is no font key for 14 point Helvetica bold on an HP LaserJet, a reduction rule will be created to substitute another font when 14 point Helvetica bold is requested. If you then get an HP LaserJet font cartridge that includes 14 point Helvetica bold and create the font key for it, the reduction rules will automatically be changed to allow that new font to be printed.

When to Add Reduction Rules

You will probably never need to create or modify reduction rules unless you want to override the automatic reduction rules. If you do override the automatic reduction rules, it is usually for aesthetic reasons, for example, if you want the replacement font to be bold instead of italic. However, there are a couple of reasons for adding reduction rules that go beyond aesthetics, which are discussed below.

You might need to add a reduction rule is if a replacement font family can't be derived from the sequence of similar font families. (Remember that you specify the similar font family in the Font Families directory.) For example, if the last font family in the sequence of similar font families is Courier, and your printer can't print Courier, you need to create a reduction rule to substitute a font family that can be printed.

You may also need to add reduction rules to omit font keys from the run-time font database without deleting them from the editable font database.

Note: *The rest of this section discusses how reduction rules work, how they are created automatically, and how the Font Service reads them. Since you may never need to add or modify reduction rules, you might want to skip ahead to the next section. However, if you do need to add reduction rules, read the rest of this section.*

Sample Reduction Rules

The Reduction Table, shown in Figure 2-25, lists the font substitution rules. When you add or modify reduction rules, you work with the Reduction Table.

REDUCTION TABLE

Family	Points	Bold	Ital	Device Type
*	19–23	*	*	Imagen
*	18	*	*	Imagen
Comment: 18 point is the 3rd largest point size available on the IP/III				
*	24–35	*	*	Imagen
*	24	*	*	Imagen
Comment: 24 point is the 2nd largest point size available on the IP/III				
Greek	25–99	*	*	Imagen
Greek	24	*	*	Imagen
Comment: 24 point is largest Greek on Imagen				
Cubic PS 96	*–*	*	*	
Cubic PS 96	*	*	*	Daisy
Comment: Default device is Daisy when family is Cubic PS 96				
Helvetica	*–*	*	*	Daisy
Cubic PS 96	*	*	*	Daisy
Comment: Use Cubic PS 96 in place of Helvetica on Daisy				

501A.2–25

Figure 2–25. The Reduction Table

Figure 2-26, below, shows a sample reduction rule. The requested font is 32 point Times roman, and the replacement font is 36 point Times roman.

In this figure, the first line of a reduction rule indicates the fonts the rule applies to. In this case, the reduction rule only applies to one font: 32 point Times roman.

The second line specifies the replacement attributes. In this case, only the point size will be replaced.

The third line of the reduction rule explains the reason for the rule. In this example, the comment indicates that the printer can't print 32 point Times roman.

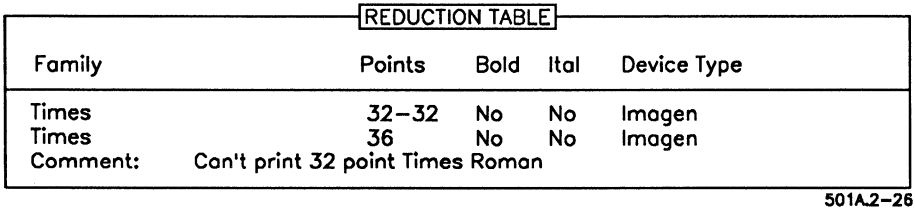


Figure 2–26. Sample Reduction Rule

Figure 2-27 shows an example of another reduction rule. This reduction rule uses asterisks, or *wild cards*. Wild cards in the first line accept any requested value. In this example, there are wild cards in the *Points, Bold,* and *Ital* fields, which means that the rule applies to all requested Helvetica Daisy fonts of any size and of any type style.

The second line defines the limitations that apply. In this example, the only font attribute changed is the font family, which changes Helvetica to Cubic PS 96. The purpose of this rule, therefore, is to substitute the font family Cubic PS 96 whenever the font family Helvetica is requested from a daisy printer.

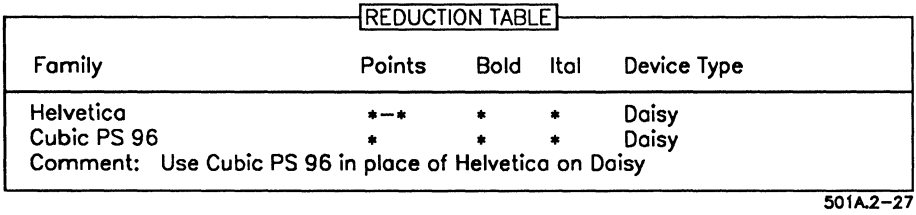


Figure 2–27. Sample Reduction Rule With Wild Cards

How Reduction Rules Are Created

There are two kinds of reduction rules: automatic and visible. Automatic reduction rules are created automatically; they do not appear in the Reduction Table.

Visible reduction rules are the rules displayed in the Reduction Table. Several visible reduction rules come with the standard Font Database; they are designed to enforce font substitutions that wouldn't be carried out automatically.

When you create reduction rules, you create additional visible reduction rules. The procedure for creating visible reduction rules is described in Section 3.

Automatic reduction rules are created using the data in the font keys. For example, if there are no HP LaserJet Lubalin font keys, a reduction rule will be created to substitute a different font family when Lubalin is requested for an HP LaserJet.

Automatically created reduction rules have a hierarchy governing the font attribute that will be substituted. When you choose a replacement font, the first attribute changed by the reduction rules is the type style. If a replacement font can't be found by changing the type style, then the reduction rules change the point size. If a replacement font can't be found by changing the point size, then the reduction rules change the font family. Here is an example.

Let's say a user requests 14 point Helvetica bold. There is no font key for it, however, so a replacement font must be found. There are three replacement fonts available:

1. 14 point Times bold
2. 12 point Helvetica bold
3. 14 point Helvetica roman

In this case, the replacement font will be 14 point Helvetica roman. Rather than change the font family or the point size, the reduction rules will change the type style.

Whenever a font attribute is substituted, the reduction rules automatically select the closest value possible. For example, if the available point sizes are 12, 10, and 8, 13 point type will be replaced by 12 point type. Substituting type styles is a bit more complex, since there are four possibilities. The following table shows the substitution order for each missing type style. For example, the table shows that if bold italic is not available, the first choice substitute is italic, the second choice is bold, and the third choice is roman.

Missing Style	1st Choice	2nd Choice	3rd Choice
Bold Italic	Italic	Bold	Roman
Bold	Roman	Bold Italic	Italic
Italic	Bold Italic	Roman	Bold
Roman	Bold	Italic	Bold Italic

Proper Rule Order

When you work with reduction rules, you need to understand how the Font Service reads the Reduction Table. The Font Service starts at the top of the Reduction Table and works down until it comes to a rule that applies to the requested font. The Font Service then goes back to the top of the reduction table and works down again until it comes to another rule that applies. Each time it goes through the rules, the Font Service ignores the rules that have already been used.

Figure 2-28 shows an example of how a requested font can have two different replacement fonts depending on the order of the entries in the reduction table. In rule order "A," 24 point Courier will be replaced by 12 point Courier. If you reverse the order of the entries in the reduction table, as in rule order "B," 24 point Courier will be replaced by 14 point Helvetica bold. This will happen because both rules can be applied to the requested font. The rule that is applied first is the one that defines the replacement font.

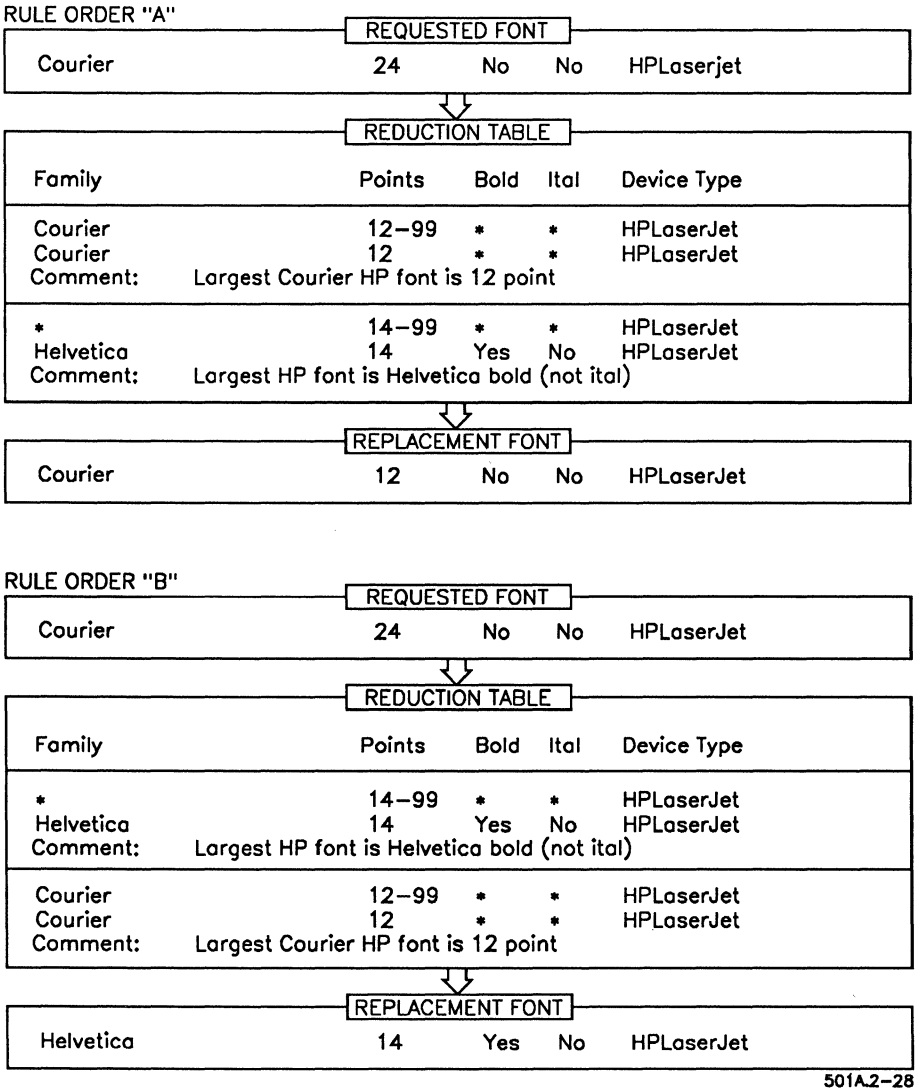

RULE ORDER "A"
REQUESTED FONT
Courier 24 No No HPLaserjet
REDUCTION TABLE
Family Points Bold Ital Device Type
Courier 12–99 * * HPLaserJet
Courier 12 * * HPLaserJet
Comment: Largest Courier HP font is 12 point
* 14–99 * * HPLaserJet
Helvetica 14 Yes No HPLaserJet
Comment: Largest HP font is Helvetica bold (not ital)
REPLACEMENT FONT
Courier 12 No No HPLaserJet
RULE ORDER "B"
REQUESTED FONT
Courier 24 No No HPLaserJet
REDUCTION TABLE
Family Points Bold Ital Device Type
* 14–99 * * HPLaserJet
Helvetica 14 Yes No HPLaserJet
Comment: Largest HP font is Helvetica bold (not ital)
Courier 12–99 * * HPLaserJet
Courier 12 * * HPLaserJet
Comment: Largest Courier HP font is 12 point
REPLACEMENT FONT
Helvetica 14 Yes No HPLaserJet
501A2-28

Figure 2–28. Two Reduction Rule Orders

There is one more point to made about rule order. Visible reduction rules are always at the top of the table. This makes it easier for you to override the automatic reduction rules, since the Font Service reads the rules that you create first.

Nil Values

As mentioned earlier, the default font family is used when no font family is specified. In applications other than Document Designer, you can create a nil value reduction rule to override the default font family.

Figure 2-29 shows a nil value reduction rule. It specifies that when no font family is named, Draft will be used. (Remember that in Document Designer, the default family will still be used.) This rule creates two default font families, one for Document Designer, and one for the rest of your applications, for example, Extended Multiplan. (This rule is primarily designed for users who prefer to print spreadsheets in Draft, which prints faster than other fonts.)

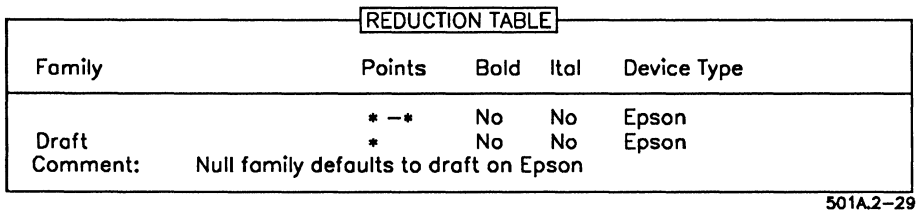


Figure 2–29. Nil Value Reduction Rule

Font Tool Commands

In addition to the directories on the Font Tool main menu discussed earlier in this section, the main menu includes four commands, which are described below.

Generate Runtime Database

Any time you edit the editable font database, you need to generate a new runtime font database that incorporates the revisions. To do so, you use the Generate Runtime Database command. Once you generate a new runtime font database, you need to copy it from the Font Tool directory *<FontDb>* to the directory that contains your old runtime font database, usually *<Gps>*. (For more information, see "Generating a New Runtime Font Database," in Section 3.)

Convert Wheel Set File

The Convert Wheel Set File command allows you to convert Word Processor wheel set files so they can be used by the Font Service. This command uses the data in the files to automatically create new translation tables, character sets, and width tables. The procedure is given in Section 3.

Import and Export

Many of the files in the editable font database are cross-referenced with each other. When you move or copy any files, they must be moved or copied in such a way that their cross-references stay intact. The Import and Export commands allow you to send and receive data between editable font databases. The Export command lets you to choose the data you want to send. The Import command is used to receive the incoming data. The Export Variant command works the same way as the Export command, but it only exports variant data.

The only files in the editable font database that you can copy or move without using the Import or Export commands are the font files in the Screenfont Database. These are the files with the suffix *.font*. You can copy these files using the Executive Copy and LCopy commands.

Batch Mode Commands

You usually use the batch mode commands when you want to distribute a custom version of an editable font database with distribution diskettes, for example, as part of the initial GPS installation procedure. You also use these commands when working with variant databases. The batch code commands are described in "Batch Mode Commands," in Section 3.

How Applications Use Font Data

Section 1 presented an overview of how Document Designer and OFIS Designer access font data from the runtime font database. The first part of this section dealt with each individual type of font data. The following example is specific to Document Designer and shows how each part of the font data is used to get Document Designer text from your system to the printed page.

Scenario

There has been some confusion in your office lately regarding the definition of turnips and parsnips, and you have been asked to create a short, professional looking document that sums up and clarifies the arguments. You'll use Document Designer to create the document and an Imagen laser printer to print it.

You open your document and set the font on the first paragraph to 13 point Helvetica roman. You notice that the format status line at the top of your screen shows 12 (13) point Helvetica. That's close enough, so you enter the following statement:

Turnips ≠ Parsnips

It looks fine on your screen, so you print it. Now let's look at how each element of the Font Database contributed to getting that statement from your keyboard to the printed page.

The Font Families directory: Since Helvetica is included in the Font Families directory, the name *Helvetica* is displayed in the Document Designer Fonts menu. This allows you to choose Helvetica as the font family.

The Device Types directory: Since Imagen is included in the Device Types directory, you can print to the Imagen printer. Since the video120 device is included in the Device Types directory, you can see the text on your screen.

Translation tables: In all, six translation tables are used to get your final printed document: two for the text, two for the not equal (≠) character, and two for the raster fonts.

Character sets: In all, three character sets are used: one for the text, one for the not equal (≠) character, and one for the raster fonts.

Width tables: Two width tables are used: one for the not equal character (*ImagenSymbol12*), and one for the rest of the text (*IThelvr12*).

Font keys: Two font keys are used: one for the printer font, and one for the raster font.

Reduction rules: Since there are no 13 point Imagen fonts, reduction rules are created to substitute 12 point instead. You see the substitution in the format status line on your screen.

Pseudo font families: Since the document is created with Document Designer 2.2, no WP wheel sets are used, so no pseudo font families are needed.

Raster fonts: Document Designer uses the raster font file named in the raster font font key to display the text on the screen (unless you are using the Scaling Font Service).

Font Tool Commands: The only command that directly applies to this situation is the Generate Runtime Database command. To be able to print your document, someone had to generate the runtime font database that included all the font data described above.

Section 3
Step-by-Step

This section provides step-by-step instructions for all the Font Tool tasks. The background information for this section is provided in Section 2. It may also be helpful to refer to Section 4 for some concrete examples of how to use the Font Tool.

Getting Ready

There are a few guidelines and some important information you need to know before starting to work with the Font Tool. This section covers function keys, status flags, and the best ways to move around within directories and detail forms.

Guidelines for Working With the Font Tool

When you work with the Font Tool, you must often go through several procedures to accomplish just one task. For example, to add a font, you might have to add a font family, a width table, a translation table, a character set, a font key, and then change a few reduction rules. Yet, to add another font, you may only need to add one font key. Therefore, it is difficult to make definite statements about the steps needed to complete a particular task, for example, to add font families or device types.

Generally, you'll perform the following steps, in this order, to add font data to the Font Database:

- create a variant database
- add new font families
- add new device types
- add width tables
- add new character sets and translation tables
- add font keys
- add or modify reduction rules, if necessary
- generate new runtime font database

Getting Around in the Font Tool

To help you navigate efficiently through the various tables and directories in the Font Tool, follow the guidelines below.

- Use the **SCROLL** keys or the **NEXT PAGE/PREVIOUS PAGE** keys to move up or down.
- Use **CODE-FUNCTION** keys to jump through directories. For example, **CODE-F4** takes you 40 percent of the way through a directory, and **CODE-F10** takes you to the end.
- Use **TAB** and **CODE-TAB** or **NEXT** and **CODE-NEXT** to move from field to field in detail forms.
- Use **CODE-DELETE** to clear a field.
- Use **MARK**, **BOUND**, and **MOVE** to move records. You only need to move records when working with the Reduction Table and the Font Families and Device Types directories. All other directories are reorganized alphabetically every time you open the Font Tool.

Edit Mode and View Mode

When you first enter the Font Tool you are in view mode, which means that you can look at records, but you can't make changes to them. In order to modify or add font data, you need to be in edit mode. When you are in edit mode, you will see a blinking cursor, which indicates that it's your turn to enter or alter data. There are a number of ways to enter edit mode, as explained below in the section "Special Keys." Most of the time, when you're in edit mode, you'll be working with detail forms. This is not the case, however, when working with reduction rules and records in the font families and device types directories. These records have no detail forms, so when you're in edit mode, you'll be modifying the records directly.

Special Keys

The following keys in the Font Tool have unique functions:

F1 (Redo or Top)

Most of the time, **F1** functions as the **TOP** key, which takes you to the beginning of a directory.

In edit mode, however, **F1** changes from **TOP** to **REDO**. The **REDO** key functions as you might expect; it reenters what you last typed in a field. It also repeats the keystroke used to finish the entry in a record. For example, if you wanted to enter No in the *Gen?* field in several records in the Font Families directory, you would perform the following steps:

1. Highlight the record to modify and press **F5**.
2. Press **TAB** to move the highlight to the *Gen?* field, delete the Yes, and enter **No**.
3. Press **GO** to finish the entry.

To set the *Gen?* field to *No* on the next entry, you would use **F1**, as follows:

1. Highlight the record to modify and press **F5**.
2. Press Tab to move the highlight to the *Gen?* field and press **F1**. Pressing **F1** deletes the Yes, enters **No**, and repeats the **GO** key. You are then ready to modify the next record.

F2 (Revert)

Modified records in a variant database are flagged with a plus sign (+). If you want to restore a record to its original state, highlight the record and press **F2** (Revert). The plus sign disappears, and the data in the entry reverts back to the original database.

If you have created a new record, it only exists in the variant database, so pressing **F2** will not place it into the original database. Instead, you will just hear a beep.

If you delete an original record while working in the variant database, you can restore it by recreating the record and pressing **F2**. The record is restored to the original database.

Note that you don't use this key when working with the Font Families and Device Types directories or the Reduction Table. This is to insure that the order of these directories is kept intact, as explained further in the section "Status Flags," later in this section.

F3 (Copy)

Places you in edit mode and creates a copy of the highlighted record or detail form which you can then modify.

F4 (Insert)

Places you in edit mode and creates a blank record or a blank detail form where you can enter new data.

F5 (Modify)	Places you in edit mode and displays a detail form of the highlighted record which you can then modify.
F6 (Delete)	Deletes highlighted records and fields. In general, try not to delete data from the Font Tool, unless it is obviously redundant or invalid. Font data that seems useless now may be necessary if you get a new printer at a later date.
Help	If a field requires a name from another directory, pressing **HELP** displays that directory. For example, if you are in the *Font Families* field of the Reduction Table, you can press **HELP** to access the Font Families directory. You can then copy a name from the Font Families directory back into the Reduction Table by highlighting the name and pressing **GO**.
Code-Help	Use this as a shortcut when you want to go from one detail form to another. For example, if you are in the detail form of a font key, and you need to make a change to the width table, just highlight the *Width Table* field in the font key, and press **CODE-HELP**. This bypasses the Width Tables directory and takes you directly to the width table that you want to edit.

There are additional function keys that are only displayed when you work with a particular directory or detail form. These function keys will be discussed in their appropriate procedures.

Status Flags

Status flags inform you of the status of each record in the editable font database. There are three types of flags: bullets, daggers, and plus signs. These flags can be displayed in three different ways: bright, dim, or outline/blinking. Usually you use status flags for record keeping. The only exceptions are the outline or blinking status flag, as explained below.

Bullet (•)	Indicates an automatically created record. For example, if you add a device type and don't create any font keys for it, font keys pertaining to that device type are generated automatically when you generate the runtime font database. Also, a bullet may indicate a record that has been created by the Convert Wheel Set command, for example, a translation table.
Dagger (†)	Indicates a user-created record. It also indicates that the record was not included the last time the runtime font database was generated.
Plus Sign (+)	Indicates that the record is part of a variant database. Any time you enter the Font Tool in a variant font database and modify or add a record, this flag will be displayed with that record. In this way, you can keep track of your changes. In the Device Types and Font Families directories, the plus sign functions slightly differently. When you add a new device type or font family to a variant database, you won't see a plus sign. However, when you exit the Font Tool and return to the directory, every record will be flagged with a plus sign. This is because the order of the records is often important in these directories, so when you add a record, all of them are included in the variant database. In this way, the order is kept intact.
Bright	A bullet is bright when a record is first created or when a record was included in the runtime font database the last time it was generated. A plus sign is bright if a record was included in the runtime font database. There are no bright daggers.

Dim	A dim status flag of any kind means that a record was not included in the runtime font database the last time it was generated.
Outline or Blinking Status Flag	An outline status flag (on high-resolution monitors) or a blinking status flag (on all other monitors) indicates that a record is invalid. For example, a font key may contain the name of a width table that has been deleted. Usually the invalid field is highlighted; but in any case, you need to find what is invalid about the record and correct it.

Creating a Variant Database

To use a variant database, you must first create it. (For information on variant databases, see "Variant Font Databases," in Section 2.)

To create a variant database,

1. Path to *[Sys]<FontDb>*.
2. Type **Font Tool**
3. Press **RETURN.**
4. The Font Tool command form is displayed, as shown below.
5. Type **Create** in the *Batch mode commands* field.
6. In the *Variant Database* field, enter the name of your variant database. The name should indicate the modifications you are about to make. For example, if you are creating a font database for Angola, enter **Angola**, as shown below.

```
Font Tool
  [Batch mode commands]      Create
  [Variant database]         Angola
```

7. Press **GO**. It takes a few minutes to create the empty files that will contain your modified or added records. When the command line returns to the screen, you can enter the Font Tool.

Entering the Font Tool

You can enter the Font Tool through the Executive or through the Context Manager. If you enter through the Executive, you can enter either the original database or a variant database. If you enter through the Context Manager, however, you can only enter into the original database.

If you enter through the Executive, you must first path to *[Sys]<FontDb>*. This directory was created when you installed the Font Tool.

To enter the Font Tool through the Executive,

1. Path to *[Sys]<FontDb>*.
2. Type **Font Tool**
3. Choose one of the following:
   a. If you want to work with the original database, press **GO**.
   b. If you want to use a variant database, press **RETURN**. Enter the name of the variant database in the *Variant Database* field, and press **GO**. If you get an error message, it means you haven't created your variant database yet. (See the previous subsection, "Creating a Variant Database.") The example below shows how to enter the variant database *Angola*.

```
Font Tool
   [Batch mode commands] _____
   [Variant database]    Angola_____
```

You may run into a couple of problems at this point, whether or not you use a variant database.

- If, instead of the main menu, you get an error code and a status message that states that certain files are missing, you might have missed a step during GPS installation. Refer to the Release Notice and try the FontDb or ScreenfontDb Submit command.
- If you open the Font Tool and find that all the directories are empty, it means that you didn't path to *<FontDb>* before issuing the Font Tool command. Press **FINISH** and begin again from step 1.

The Main Menu

When you open the Font Tool, you first see the main menu, as shown in Figure 3-1. (Scroll up to see the last three fields in the menu.) The name of the variant database is displayed at the top of the menu.

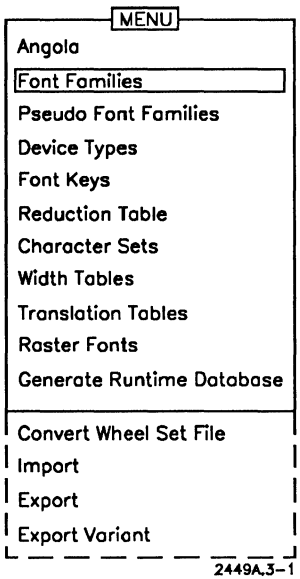


Figure 3–1. The Main Menu

To access a directory or execute a command, use the arrow keys to move the highlight to the appropriate field, and press **GO**. To return to the main menu from elsewhere in the Font Tool, press **CANCEL**.

The last entries on the menu are Font Tool commands. (If you are in the original database, there is no Export Variant command.)

Note: *If, as you're working through the material in this section, you decide to generate a new runtime font database by issuing the Generate Runtime Database command, the only way to halt the command is to press* **ACTION-FINISH**. *The command is halted, and you are returned to the Executive. Otherwise, if you wait for the command to finish executing, it could 15 minutes or longer.*

Working With Font Families

You work with the Font Families directory whenever you add font families to or remove families from the runtime font database. If the font family is not included in the editable font database, you need to add a record. If the font family is included in the Font Families directory but is not displayed in the Document Designer *Fonts* menu, you need to modify the font family's record to include it in the runtime font database. For more information on font families, see "Font Families," in Section 2.

Modifying a Font Family Record

There are two fields in the Font Families directory that you can modify: *Gen?*, which determines if the font family is in the runtime font database; and *Similar Font Family*, which gives the first choice for font substitution.

To modify a record in one of these fields,

1. Access the Font Families directory from the main menu. (The Font Families directory is shown in Figure 3-2.)
2. Use the arrow keys to move the cursor to the record you want to modify. Press **NEXT PAGE** to see more records.

3. Press **F5** (Modify).
4. Use **TAB** and **CODE-TAB** to move the highlight through the record.

   Enter **yes** or **no** in the *Gen?* field. Enter **yes** if you want the font family in the runtime font database; enter **no** if you do not want the font family in the runtime font database.

   If you enter **no** in the *Gen?* field for the first font family, it will no longer be the default font family. Instead, the default font family will be the first one in the list that has *Yes* in the *Gen?* field.
5. Enter the name of a font family in the *Similar Font Family* field. This is your first choice for font substitution; it must be another font family in the directory. Remember that it's a good idea to choose a font family that is closer to the top of the directory.
6. Press **GO** to save the changes; press **CANCEL** to discard them. If you want to modify more records, repeat the procedure from step 3 above. Otherwise, press **GO** to return to the main menu.

FONT FAMILIES

Font Family	Gen?	Proportional?	Similar Font Family
Courier	Yes	No	
Helvetica	Yes	Yes	
Times	Yes	Yes	
Cubic PS 96	Yes	Yes	
CourierWP	Yes	No	
Greek	Yes	Yes	
Draft	Yes	No	Courier
Palatino	Yes	Yes	Times
ITC Avant Garde Gothic	Yes	Yes	Helvetica
ITC New Century Schoolbook	Yes	Yes	Times
Helvetica Narrow	Yes	Yes	Helvetica Narrow
ITC Bookman	Yes	Yes	New Century Schoolbook
ITC Zapf Chancery	Yes	Yes	Times
ITC Zapf Dingbats	Yes	Yes	
ITC Lubalin Graph	No	Yes	Courier
ITC Souvenir	No	Yes	ITC Bookman

501A.3-2

Figure 3–2. Font Families Directory

Adding a Font Family Record

Remember, to avoid unnecessary or redundant data, it is a good idea to limit the number of font families in the Font Families directory. Before you add a new font family, make sure that it isn't just a variation of an existing font family. For example, Elite is another name for 10 point Courier.

To add a font family,

1. Access the Font Families directory from the main menu. (The Font Families directory is shown in Figure 3-2.)
2. If you want the new font family to be the default, leave the highlight at the top of the directory. Otherwise, press **DOWN ARROW** to move the highlight down through the directory. Remember that the default font family must be monospaced.
3. Press **F4** (Insert) to display a new record. Use **TAB** and **CODE-TAB** to move the highlight through the record.
4. Enter the name of the new font family in the *Font Family* field.
5. Enter **yes** or **no** in the *Gen?* field.

   If you enter **no** in the *Gen?* field for the first font family, it will not be the default font family. Instead, the default font family will be the first one in the list that has the Gen? field set to *Yes*.
6. Enter **yes** in the *Proportional?* field if the font is proportional; enter **no** if the font is monospaced.
7. Enter the name of a font family in the *Similar Font Family* field. This is your first choice for font substitution; it must be another font family in the directory. Be careful to avoid setting up circular similar font families, for example, where Helvetica is substituted for Courier, and Courier is substituted for Helvetica. The best way around this problem is to choose a similar font family from the font families listed above the one you are adding.
8. Press **GO** to save the record; press **CANCEL** to discard it. If you have any more font families to add, repeat the procedure from step 3 above. Otherwise, press **GO** to return to the main menu.

Once you have added the new font family, there are probably several more procedures to complete; for example, you may need to add translation tables, width tables (if the new font family is proportional), and font keys.

Working With Device Types

You use the Device Types directory to add device types to or remove them from the runtime font database. For example, you might have to add a device type if you get a new printer. You also use this directory to specify the default device type. For information on device types, see "Device Types," in Section 2.

DEVICE TYPES

Device Type	Gen?	Resolution Hor.	Resolution Ver.	Print? Rasters?	Print?	Print? Alias?	Scan?
Imagen	Yes	*	*	No	Yes	Yes	No
Daisy	Yes	*	*	No	Yes	Yes	No
EpFx286	Yes	*	*	No	Yes	Yes	No
PostScript	Yes	*	*	No	Yes	Yes	No
HPLaserJet	Yes	*	*	No	Yes	Yes	No
Simple	No	*	*	No	Yes	No	No
LaserWriter+	No	*	*	No	Yes	Yes	No
Video120	No	120	120	Yes	No	No	No

501A.3-3

Figure 3–3. Device Types Directory

Modifying a Record in the Device Types Directory

You can modify a record in the Device Types directory to include it in or exclude it from the runtime font database. To modify a record,

1. Access the Device Types directory from the main menu. (Figure 3-3 shows the Device Types directory.)
2. Use the arrow keys to move the cursor to the record you want to modify.

3. Press **F5** (Modify). The only field you can modify is the *Gen?* field. Enter **yes** if you want to include the device type in the runtime font database; enter **no** if you do not.

   Remember that the first device type is the default device type. If you enter **no** in the *Gen?* field for the first device type, it will no longer be the default. Instead, the default device type will be the first device type in the list that has *Yes* in the *Gen?* field.

4. Press **GO** to save the changes; press **CANCEL** to discard them. If you want to modify any other records, repeat the procedure from step 3 above. If not, press **GO** to return to the main menu.

Changing the Default Device Type

You want the default device type to correspond to the printer you use most. To change the default device type, you need to move the device type that you want to be the default to the top of the Device Types directory. To do this,

1. Access the Device Types directory from the main menu.
2. Use the arrow keys to move the highlight to the desired device type.
3. Press **MARK**. Move the highlight to the top of the list, and press **MOVE**. The marked record moves to the top of the directory; it is now the default device type.

If you change your default device type, you should change the default printer entry in your user file. For information on how to do so, see the *Document Designer: System Administration Manual.*

Adding a Device Type to the Device Types Directory

When you get a new kind of printer, you'll probably need to add a device type to the Device Types directory. Remember that device types are defined by printer technology and by font capability. If you have two HP LaserJet printers that each print different fonts, you may want to create a device type for each of them. If you have written a device driver for a new printer, then you definitely need to add a device type.

If you don't see the name of your printer in the Device Types directory, it may still be included in the Font Database. For example, the Diablo 630 API-2 is supported by the daisy device type, along with several other daisy wheel printers. For more information, see "Device Types in the Standard Editable Font Database," in Section 2.

Remember that when you add a device type, you won't be able to use it until you install it (using the Print Manager). For information on installing printers, see the *CTOS Generic Print System Administration Guide*.

To add a device type to the Device Types directory,

1. Access the Device Types directory from the main menu.
2. If you want the new device type to be the default, leave the highlight at the top of the directory. Otherwise, press the **DOWN ARROW** key once.
3. Press **F4** (Insert) to create a new record. (Use **TAB** and **CODE-TAB** to move the highlight through the record.)
4. Enter the name of the new device type in the *Device Type* field.
5. Enter **yes** in the *Gen?* field if you want the device type to be included in the runtime font database; enter **no** if you don't.

   If you enter **no** in the *Gen?* field for the first device type, it will not be the default. Instead, the default device type will be the first one in the list where the *Gen?* field is set to *Yes*.
6. If you are adding a device type for a printer, enter an asterisk (*) in both the *Hor* (Horizontal) and *Ver* (Vertical) fields. If you are adding a video device type, enter **120** in both fields (or whatever the resolution of the monitor, in pixels).
7. If the device type is a printer, enter **no** in the *Rasters?* field; if the device type is a monitor, enter **yes**.
8. If the device type is a printer, enter **yes** in the *Print?* field; if the device type is a monitor, enter **no**.
9. If the device driver requires aliases, enter **yes** in the *Alias* field. For information on alias requirements, see Appendix B and your printer documentation.

10. In the *Scan?* field, enter **no** if the device type does not use downloadable fonts.

    If the device type does use downloadable fonts, choose one of the following:

    a. If the device driver needs to scan for downloaded fonts before printing, enter **yes**.

    b. If you are not certain if the device driver needs to scan for downloaded fonts, enter **no**. With *No* as the value in the *Scan?* field, when you print, you may find that a new page starts with every downloaded font. If that is the case, you need to enter **yes** in the *Scan?* field.

11. Press **GO** to save the record; press **CANCEL** to discard it. If you want to add another device type, repeat the procedure from step 4 above. Otherwise, press **GO** to return to the main menu.

12. If you created a device type for printing downloadable fonts, you should probably store your downloadable font files in *[Sys]<Gps>*. If you store them in a different volume or directory, you need to specify the volume and directory in the *Device Setup* field in the *Device* form when you use the Print Manager to install the printer. In this way, the device driver can find the location of the volume and directory of the downloadable fonts. In the *Device Setup* field, enter

    dir:[*VolumeName*]<*DirectoryName*>

    where *VolumeName* and *DirectoryName* are the names of the volume and directory, respectively, where you want to store the downloadable font files. (Make sure you enter both the volume name and the directory name.)

    If you don't enter this parameter, the device driver searches in *[Sys]<Gps>* for the downloadable font files. For more information, see the *CTOS Generic Print System Administration Guide*.

Working With Translation Tables

When you add a new device type, you'll usually need to create a new translation table. By doing so, you add the device type's character code set to the Font Database, which ensures that the character codes used by your application programs are translated into the character codes used by the device type.

You need to modify translation tables in order to change character code translations or to access characters that are on the printer but not included in the CTOS character code set. For more information on translation tables, see "Translation Tables," in Section 2.

When you work with translation tables, the following rules apply:

- In the Font Tool, character codes are represented by hexadecimal codes, as opposed to decimal numbers.
- Each hexadecimal code must be made up of two characters, for example, 1Bh, 67h, FFh, and so forth. (Remember that the codes in this manual are suffixed with an *h* to distinguish them from decimal numbers.)
- Strings of hexadecimal codes cannot be longer than 8 codes, for example, A1A2A3A4A5A6A7A8h.
- There must be an even number of characters in each string of hexadecimal codes.
- Hexadecimal values below 20h are escapes. (Exceptions to this rule occur with the 01 and 02 escapes. See Appendix E for information on translation table escapes.)
- Some hexadecimal codes have specific meaning for the Document Designer application. For more information, see Appendix F.

Modifying a Translation Table

To modify a translation table,

1. From the main menu, access the Translation Tables directory, as shown in Figure 3-4.

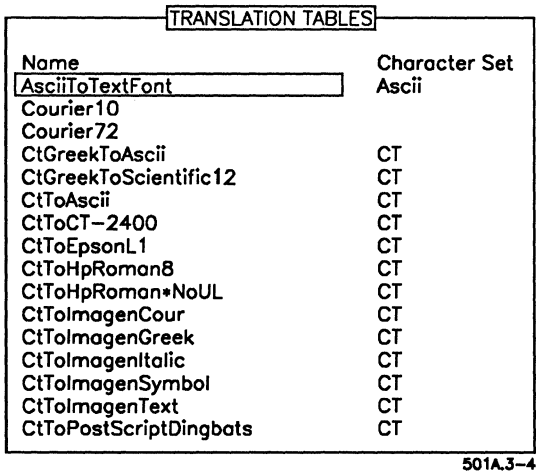


Figure 3–4. Translation Tables Directory

2. Highlight the name of the translation table you want to modify. (Use **TAB** and **CODE-TAB** to move the highlight through the table.) Press **F5** to display the translation table.
3. Enter any explanatory information in the *Comment*: field. This is an optional field that you can use to clarify the purpose of the table.
4. Change the character codes or add codes as necessary. If you are adding codes, try to add them in unmapped locations so that you don't delete existing characters.

   Remember that your printer documentation may give the character code in decimal format. See Appendix C for a decimal/hexadecimal conversion table.
5. Press **GO** to save your changes; press **CANCEL** to discard them.

Creating a New Translation Table

If you add a new device type, you'll probably need to create a new translation table. You may also have to add a new translation table if you add a symbol font that is not included in the editable font database, for example, the HP LaserJet Linedraw font.

To add a new translation table,

1. From the main menu, access the Translation Tables directory.
2. Press **F4** (Insert). A blank translation table is displayed, as shown in Figure 3-5. (Use **TAB** or **CODE-TAB** to move the highlight through the table.)

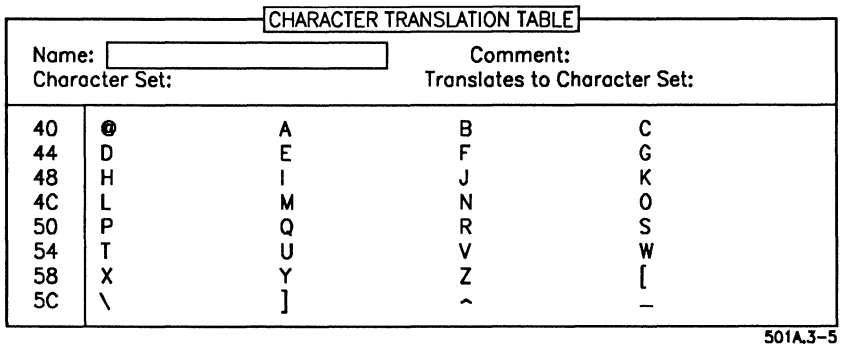


Figure 3–5. Blank Translation Table

3. Enter a name in the *Name*: field. The name should reflect the purpose of the table. If it's a first level translation table, use the format **CTto***Devicetype*, where *Devicetype* is the name of a device type, for example, **CTtoPostScriptN**. If it's a second level translation table, enter the name of the device type and the name of the type of code, for example, **ImagenTextToGascii**.
4. Enter comments as needed in the *Comment*: field. This is an optional field that you can use to clarify the function of the table.

5. In the *Character Set*: field, enter the character code set you are translating from. For first level translation tables, this is usually the CTOS character code set, in which case you enter **CT**. For second level translation tables, use the name of the character code set of the device type.

6. In the *Translates to Character Set:* field, enter the name of the character set that uses this translation table. If necessary, use **HELP** to access the Character Sets directory. Chances are you haven't created the character set yet, in which case you should do so now. To create the character set,

   a. Highlight the *Translates to Character Set:* field, and press **HELP** to access the Character Sets directory.

   b. Press **F4** (Insert) to create a blank character set.

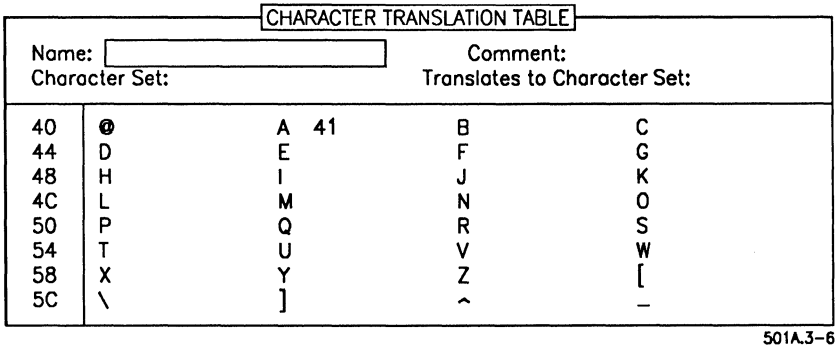
   c. In the *Name:* field, enter the name of the new character set. It should reflect the device type that the character set will be used by, for example, *PostScriptN*. For now, leave the rest of the fields blank; you will fill them in later.

   d. Press **GO** to enter the name of the new character set in the Character Sets directory.

   e. Press **GO** again to enter the name of the new character set in the *Translates to Character Set:* field in the new translation table.

7. Use your printer documentation to find the hexadecimal code that each character is mapped to in the character code set of the device type. If the printer documentation uses decimal codes, see Appendix C for a decimal/hexadecimal conversion table.

   Enter the printer's character codes to the right of the characters in the table. For example, if the printer's character code for A is 41h, enter **41** next to the A (see Figure 3-6).


CHARACTER TRANSLATION TABLE

Name:
Comment:
Character Set:
Translates to Character Set:

40	@	A 41	B	C
44	D	E	F	G
48	H	I	J	K
4C	L	M	N	O
50	P	Q	R	S
54	T	U	V	W
58	X	Y	Z	[
5C	\	]	^	_

501A.3-6

Figure 3–6. Character Codes in Translation Table

The table may not show all the characters in the new font. This is especially true if you are nationalizing translation tables. For example, there is no E umlaut in the CTOS character code set, so you won't find one in the blank translation table. In such a case, you need to map it to a location that isn't used by any other character. For more information, see Appendix F and "Nationalizing Translation Tables," in Section 4.

In order to use some characters, you may need to use escapes in the translation table. For information on escapes, see Appendix E and your printer documentation.

If you are adding a new print wheel, enter the number that indicates the spoke position of the characters. For example, if the spoke position for the dollar sign ($) is 44, enter **44** next to the dollar sign in the translation table. Spoke positions are usually included in the documentation for print wheels. If they aren't, see "Adding a Print Wheel Without Vendor Data," later in this section.

8. When you have finished entering all the character codes, press **GO** to save the table, or press **CANCEL** to discard it.

The new translation table is complete; but in order to use it, you need to reference it in the character set that you created during the above procedure. To do so, see "Working With Character Sets," later in this section.

Adding a Print Wheel Without Vendor Data

If you don't have the data for the print wheel's spoke position, you can reconstruct it with the following procedure.

1. In the Editor, create a new document called *AsciiChars*. (For information on using the Editor, see the *CTOS Editor User's Guide*.)

   When you have finished creating this document, it will contain all the characters with their codes in the CTOS character code set in the range from hexadecimal 21 to hexadecimal 7E. (See Appendix A for the CTOS character code set.) As you enter the codes, your document should look something like this:

21	22	23	24	25	26	27	28	29
!	"	#	$	%	&	'	(	)

2. When you have entered all the characters in the range 21 through 7E, exit the Editor and mount the new print wheel on the printer. Be sure that the printer has already been installed with the daisy device driver.

3. Use the Print Manager to print the file *AsciiChars* in binary mode. (To do so, see the *Using the Print Manager Manual*.) If the numbers don't print correctly, refer back to the document on the screen.

4. Enter the Font Tool, and create a new translation table. For each character on the printout, locate that character in the translation table, and enter the code next to the printed character into the space next to the character in the translation table.

5. Add all the other font data to support the new print wheel, as necessary: font family, device type, font keys, and so forth.

Invert and Shuffle Commands

The Invert command (**F9**) inverts a translation table. For example, if you invert *CtToImagenText*, it becomes *ImagenTextToCt*. This is a handy way to compare the character codes assigned to a particular character in different character code sets.

The Shuffle command (**F10**) reorganizes a translation table to reflect another character code set. Use this command if you want to base a translation table on the ordering of a particular character code set. This command is useful when nationalizing translation tables. For an example, see "Nationalizing Translation Tables," in Section 4.

Working With Character Sets

You work with character sets when you add a new printer that is not supported by GPS, or when you add a symbol font that is not included in the editable font database, for example, the HP LaserJet Linedraw font.

Since character sets are simply bundles of translation tables, you need to add a character set whenever you add a translation table. It is a good idea to create translation tables *before* you create character sets.

Creating a New Character Set

To create a new character set,

1. Access the Character Sets directory from the main menu.
2. Press **F4** (Insert) to create a new character set detail form, as shown in Figure 3-7. (Use **TAB** and **CODE-TAB** to move the highlight through the detail form.)

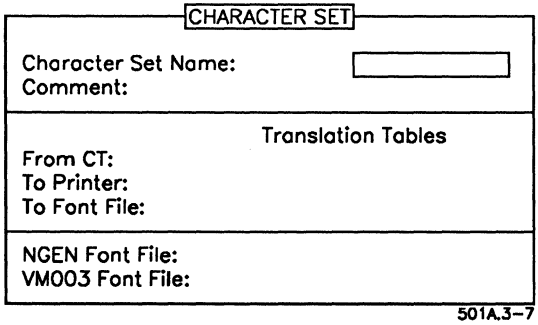


Figure 3–7. Blank Character Set Detail Form

3. Enter a name in the *Character Set Name:* field. It should reflect the device type that the character set will be used by, for example, *ImagenText*, or *HPRoman8*.
4. Enter any explanatory information in the *Comment:* field. This is an optional field that you can use to clarify the purpose of the character set.

5. In the *From CT:* field, enter the name of the first level translation table, the one that translates from the CTOS character code set to the device type's character code set. If necessary, use **HELP** to access the Translation Tables directory.

   If you haven't created the translation table that is to be entered in this field, you need to create it now. To do so,

   a. Highlight the *From CT:* field, and press **HELP** to access the Translation Tables directory.

   b. Press **F4** (Insert) to create a new translation table.

   c. Enter the name of the new translation table in the *Name:* field. The name should reflect the purpose of the table. If it is a first level translation table, use the format **CTto***Devicetype,* where *Devicetype* is the name of a device type, for example, **CTtoPostScriptN**. For now, just fill in the name. You will complete the table later.

   d. Press **GO** to add the name of the new translation table to the Translation Tables directory.

   e. Press **GO** again to add the name of the new translation table to the *From CT:* field.

6. In the *To Printer:* field, enter the name of the second level translation table containing the codes used by the printer. Usually, you need a second level translation table if the printer uses other than 8-bit character codes, for example, 7-bit or 16-bit, or if the printer uses escape sequences to access characters. (See your printer documentation to determine if you need a second level translation table; most of the time you won't.)

   If you have not yet created the second level translation table, follow the procedure in step 5 to create it. Name the translation table appropriately, using the name of the device type first, then a name to identify the codes, for example, *ImagenTextToGascii.*

7. In the *To Font File:* field, enter the name of the translation table that translates to the loadable font file. This is an optional field. If you are adding an Imagen font, enter **ImagenToTextFont**. If you are adding a PostScript font, enter **PostScriptToTextFont**. If you are adding an Epson font, enter **EpsonToTextFont**. For all other device types, leave this field blank.
8. If you entered a name in the *To Font File:* field in the above step, enter **TextSet.font** in the *NGEN Font File:* field and **VM003TextSet.font** in the *VM003 Font File:* field.
9. Press **GO** to save the character set; press **CANCEL** to discard it.

If you created a new translation table in the preceding procedure, you probably left most of it blank. Now is the time to complete it. To do so, see "Modifying a Translation Table," earlier in this section.

Working With Width Tables

You work with width tables every time you add a proportional printer font. For more information on width tables, see "Width Tables," in Section 2.

If the font you are adding requires a new translation table and character set, create those first. To do so, see "Working With Character Sets" and "Working With Translation Tables," earlier in this section.

Note: *Character width information is not usually included in the documentation that comes with your printer. You may need to contact your distributor or the printer manufacturer to get the data on character widths (sometimes called "spacing tables").*

To create a new width table,

1. Access the Width Tables directory from the main menu.
2. Press **F4** (Insert) to create a blank width table, as shown in Figure 3-8. (Use **TAB** and **CODE-TAB** to move the highlight through the table.)

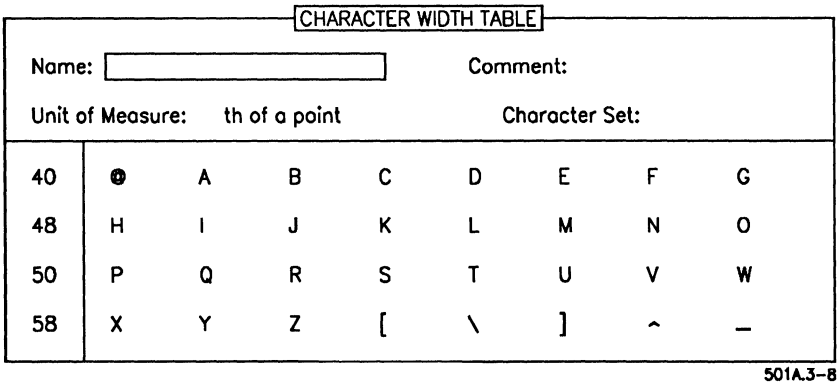

CHARACTER WIDTH TABLE
Name:
Comment:
Unit of Measure: th of a point
Character Set:
40 @ A B C D E F G
48 H I J K L M N O
50 P Q R S T U V W
58 X Y Z [\ ] ^ _
501A.3-8

Figure 3–8. Blank Width Table

3. Enter a name in the *Name:* field. It should reflect the attributes of the font. For example, *HpFHelvBold14* represents the width table for 14 point Helvetica bold printed with an HP LaserJet using an F font cartridge.
4. Enter any explanatory comment that you feel necessary in the *Comment:* field. (This is an optional field.)
5. In the *Unit of Measure:* field, enter the unit of measure in inches or in fractions of points. See your printer documentation to determine which unit of measure to use in this field. If your printer documentation gives character widths in points, it's easiest to use points as the unit of measure.

6. In the *Character Set:* field, enter the name of the character set that the font uses. For example, if you are adding a width table for an HP LaserJet text font, enter **HPRoman8**. If you are adding a width table for a PostScript text font, enter **PostScriptN**. If necessary, use **HELP** to access the Character Sets directory. For more information on how each character set is used, see Appendix D.
7. Press **SCROLL UP**. Notice that when you scroll up, the characters in the width table assume the ordering of the character set you just specified in the *Character Set*: field. This allows you to add character widths directly from the vendor documentation, since the width table is now in the printer's character code set ordering.
8. Enter the character width to the right of each character (if needed). (See your printer documentation for the character widths.)
9. Press **GO** to save the table; press **CANCEL** to discard it.

Resize, Propagate, and Shuffle Commands

The Resize command (**F8**) changes all the widths in the width table by a constant factor. For example, you can double them, triple them, or convert them to a different unit of measure.

The Propagate Widths command (**F9**) provides a short cut for filling in widths of accented characters. For example, if you fill in the width of capital *A*, this command assigns all accented capital *A* characters the same width. It does this by searching the related translation table for all compound hexadecimal codes that begin with the code for *A*, and filling in the same width for all those characters in the width table.

The Shuffle command (**F10**) rearranges the width table to reflect a different character code set. This is sometimes useful when you nationalize character sets.

Working With Font Keys

Any time you add a new font, you need to add a new font key. First, however, you must add the rest of the font data: font families, device types, translation tables, character sets, and width tables (for proportional fonts).

You may also need to modify font keys, for example, if you need to change the name of the width table that the font refers to.

Creating a New Font Key

To create a font key,

1. Access the Font Keys directory from the main menu.
2. Press **F4** (Insert) to display a new detail form, as shown in Figure 3-9.

   If you are creating a font key that is similar to an existing font key, press **F3** (Copy) to create a copy of the font key, which you can then edit.

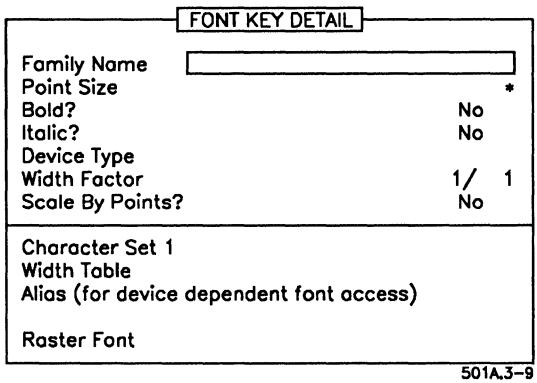


Figure 3–9. Blank Font Key Detail Form

3. Enter the name of the font family name in the *Font Family* field. It must be a name from the Font Families directory. If necessary, use **HELP** to access the Font Families directory.
4. Enter the point size in the *Point Size* field. Be sure to enter the actual point size of the font. If the font is scaled on-the-fly you will usually enter **1** (see step 7).
5. Enter **yes** or **no** in the *Bold?* field, as appropriate.
6. Enter **yes** or **no** in the *Italic?* field, as appropriate.
7. Enter the device type in the *Device Type* field. If necessary, use **HELP** to access the Device Types directory.
8. Enter the width factor in the *Width Factor* field. If it is a proportional font, leave the default set as is, *1/1*. If it is a monospaced font, enter **72** in the first part of the field, and enter the pitch of the font in the second part of the field of the form **72/***pitch*.
9. In the *Scale By Points?* field, enter **no**, unless the printer can scale fonts on-the-fly (such as a PostScript printer). If the printer cannot scale fonts on-the-fly, the Scaling Font Service will scale the font for screen display or for downloading to a supported device (such as an HP LaserJet printer). Check your printer documentation to see if your printer scales fonts.
10. Enter the primary character set in the *Character Set* field. If necessary, press **HELP** to access the Character Sets directory.
11. Enter the name of the width table in the *Width Table* field. This only applies to proportional printer fonts. If necessary, press **HELP** to access the Width Tables directory.
12. Enter any cartridge or print wheel names, escape sequences, keywords, and so forth in the *Alias* field. For information on how your device type uses the *Alias* field, see the apppropriate section in the *CTOS Generic Print System Administration Guide* and your printer or device driver documentation.
13. If you are creating a font key for a raster font, enter the name of the font file in the *Raster Font* field. For example, *HelvB09.font* is the name of the 9 point Helvetica bold raster font file distributed with the Screen Font Database. If necessary, use **HELP** to access the Raster Fonts directory.

14. If the font key uses alternate character sets, press **NEXT PAGE** and repeat steps 10 through 13. For information on alternate character sets, see "Alternate Character Sets," in Section 2.
15. Press **GO** to save the new font key; press **CANCEL** to discard it.

    If any information is invalid or missing, a message is displayed at the top of the screen. These are usually self explanatory, for example,

    No such font family

    or

    Width table is required for proportional font family.

If you have any more font keys to add, repeat the above procedure from step 2.

Modifying a Font Key

To modify a font key,

1. Access the Font Keys directory from the main menu.
2. Highlight the font key that you want to modify, and press **F5** (Modify).
3. Modify the fields as necessary. (Use **TAB** and **CODE-TAB** to move the highlight between fields.)
4. Press **GO** to save your changes; press **CANCEL** to delete them.

Tuning Raster Fonts

You can tune raster fonts so that their widths match their corresponding printer fonts. In the Screen Font Database, raster fonts are tuned to match Imagen printer fonts. If you consistently use a different type of printer, you may want to tune the raster fonts to match the fonts of that printer. Remember that when you tune raster fonts, you want to accomplish two goals:

- to get a close *overall* match between the width of the printer characters and the width of the raster font characters.
- to make the raster font aesthetically pleasing when displayed in Document Designer or OFIS Designer, avoiding characters that are too close together or too far apart.

Note: *You will not tune raster fonts if you use the Scaling Font Service, since these are scaled on-the-fly. Scaling fonts match the widths of fonts available on PostScript printers.*

When you tune raster fonts, you adjust the amount of white space around the bit map of each character, and you adjust the position of the bit map inside the surrounding white space. Figure 3-10 shows the bit map of a single raster font character. Below is a description of the elements you adjust when you tune raster fonts.

Cell Height	The height of the raster font in pixels. You can adjust the vertical position of the character within the cell height by adjusting the height of the baseline.
Baseline	The baseline of the raster font, measured in pixels, from the top of the cell.

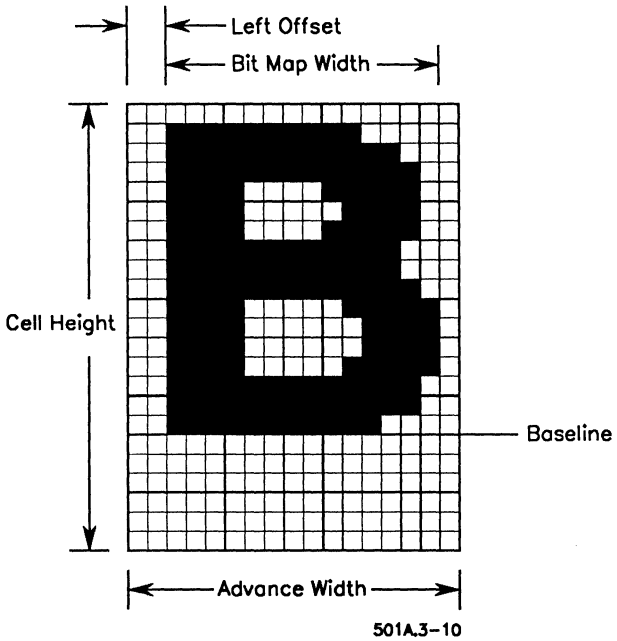


Figure 3–10. Raster Font Elements

Advance Width	The total width of the character, including the left and right offsets. By increasing or decreasing the advance width, you adjust how much space the character takes up in a line of text.
Bit Map Width	The width of the bit map for the character. You cannot adjust this width through the Font Tool. To alter the bit map, you need to use the *Raster Font Designer*.
Left Offset	The number of blank pixels to the left of the character. Use this field to move the character to the left or right.

The elements of the raster character described above are listed in the Font *Raster Detail Form*, as shown in Figure 3-11. This is the detail form you use when you tune raster fonts.

FONT RASTER DETAIL

Raster Font HelvB14.font
Character Set Video
Width Table IThelvb14
Width Factor 2/ 5
Variance * %
Cell Height: 23 Baseline: 17

	Code	Print Width	Advance Width	Bit Map Width	Left Offset
@	40	23.6	24	22	1
A	41	16.4	17	17	0
B	42	16.8	17	14	2
C	43	17.2	17	15	1
D	44	17.2	17	15	1
E	45	15.2	16	13	2
F	46	14.0	14	12	2
G	47	17.6	17	15	1

501A.3-11

Figure 3–11. Font Raster Detail Form

To tune a raster font,

1. Access the Raster Fonts directory from the main menu.
2. Highlight the raster font you want to tune, and press **F5** (Modify) to display the The *Font Raster Detail*, as shown in Figure 3-11.
3. In the *Width Table* field, enter the name of the width table that corresponds to the printer you use most often. For example, if you only use HP LaserJets, use an HP LaserJet width table. The width table must match the font attributes of the raster font. For example, if you want to tune the 9 point Helvetica bold raster font, use a 9 point Helvetica bold width table. If necessary, use **HELP** to access the Width Tables directory.

4. Make sure the width factor is correct. The width factor is a ratio used to convert the unit of measure of the printer font (points or inches) to the unit of measure of the raster font (pixels). If the printer font width table uses measurements of 300s of an inch, the width factor should be 2/5. The ratio 2/5 is a reduction of the fraction 120/300 (120 pixels per inch/300 dots per inch). In some cases, you may have to change the width factor. For example, PostScript width tables use 1000s of a point as the unit of measure. Therefore, you need to convert points to inches, as follows: 120/72,000. This fraction reduces to 1/600, but you then need to multiply this fraction times the point size, since PostScript fonts reduce to 1 point. If the font is 24 point, you end up with a width factor of 24/600, but the Font Tool can't handle a fraction that large, so you need to reduce it to 1/25.

5. Press **F8** (Widths) to display the *Variance* field. The percentage shown in the field is the variance between the raster font and the printer font. If it's a large percentage, 20% or more, it means that the raster font and the printer fonts differ significantly, or else the width factor is wrong. Be aware that the variance value is only an approximation based on one text file, so don't try to tune a font until the variance is 0.

6. Press **F10** (Show) to display the *Script* form, as shown in Figure 3-12. The *Script* form is used to compare raster and printer fonts. When a line of text is highlighted in the *Script* form, it is displayed in duplicate at the bottom of the screen. Scroll through the *Script* form to see how the text compares overall. The first line at the bottom of the screen shows how the text would print; the second line shows how the raster fonts would show the same line of text on the screen. If the lines are close to the same length, half your job is done; proceed to step 8. If the lines are different lengths, you need to work on the overall match. To do so, go to the next step.

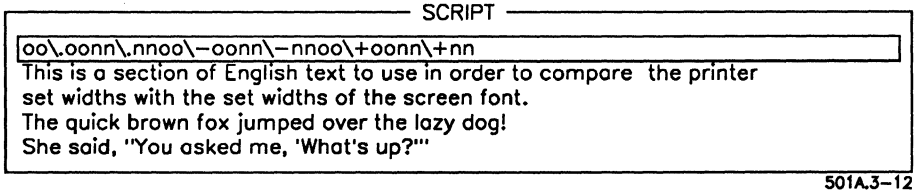

SCRIPT
oo\.oonn\.nnoo\-oonn\-nnoo\+oonn\+nn
This is a section of English text to use in order to compare the printer
set widths with the set widths of the screen font.
The quick brown fox jumped over the lazy dog!
She said, "You asked me, 'What's up?'"
501A.3-12

Figure 3–12. The Script Form

7. To get an overall match, first change the advance width of each character to match the print width. The advance width is the width of the raster font character (see Figure 3-10). The print width is the width of the character when it is printed. To do change the advance width,

   a. Press **CANCEL** to move the highlight back to the detail form.

   b. Move the highlight to the lower part of the detail form. For each character, change the number in the *Advance Width* field to match the number in the *Print Width* field as closely as possible. You can only enter whole numbers, so you need to round them off.

   c. Press **F10** (Show) to display the *Script* form. The lines of text at the bottom of the screen should be closer in length.

8. Now you need to refine the appearance of the text. You do so by adjusting the white space around the character by changing the values in the *Advance Width* and *Left Offset* fields. To add or remove space around the character, increase or decrease the advance width. To move a character to the right or left, increase or decrease the left offset. (To determine the right offset, add the left offset and the bit map width, and subtract the sum from the advance width.)

To tune individual characters, use the *Script* form, as follows:

a. Highlight the first line in the *Script* form, the line that looks like this:

```
oo\.oonn\.nnoo\–oonn\–nnoo\+oonn\+nn
```

b. Press **GO**. The highlight moves back to the detail form.

c. Move the highlight to a the *Advance Width* field in the lower part of the *Font Raster Detail* form. Notice that the slash marks in the first line of the *Script* form are replaced by the character higlighted in the detail form as well as the characters before and after it. For example, if you highlight the *Advance Width* field for the character *B*, the script form will display *A*, *B*, and *C* in place of the slash marks.

d. Check each character (for example the *A*, *B*, and *C*), to see how it pairs with the *o* and the *n* characters on either side. The *o* shows you how the character pairs with the round characters in the font; the *n* shows you how the character pairs with the straight characters. If the characters are too close or too far apart, you need to change the advance width or the left offset, or both. Remember that the advance width is the amount of white space around the character; the left offset is the amount of white space to the left of the character.

e. If necessary, change the value in the *Advance Width* field.

f. Press **TAB** to highlight the *Left Offset* field. Notice that when you press **TAB**, the line at the bottom of the screen changes to show the new advance width. If necessary, change the value in the *Left Offset* field. Press **CODE-TAB** to return to the *Advance Width* field. The line in the *Script* form changes to show the new left offset. Continue to adjust the advance width and the left offset until you get the best looking match between the characters.

As you tune each character, keep these guidelines in mind:

- Some characters look best when the bit map is centered within the advance width. To center them, subtract the bit map width from the advance width and divide the result by 2. Enter the final result in the *Left Offset* field.
- With some characters, the difference between the bit map and the advance width may be an odd number, in which case dividing by 2 gives you uneven left and right offsets. In that case, make sure that similarly shaped characters, for example *o* and *e*, have the same unbalanced offsets. In other words, if *o* has a left offset of 1 and a right offset of 2, make sure *e* has a similar left and right offset.
- Some characters require a negative value in the *Left Offset* field, for example, the italic *j*. The negative value extends the bit map out of the character cell and into the character cell of the character to the left to avoid having too much white space in front of the character.
- Some characters require that both the left and right offsets have negative values, for example, the italic *f*. (In other words, the bit map width is larger than the advance width.) Again, this is done to avoid having too much white space around the character.
- You can add your own text to the *Script* form to compare characters. Any changes you make directly to the *Script* form are not saved. To enter text permanently in the *Script* form, use the Editor to edit the file *<FontDb>Script*. For information on using the Editor, see the *Editor Manual*.

9. When you are through tuning the raster fonts, press **F9** (Update).
10. Press **GO** to save your changes; press **CANCEL** to discard them.

Working With Reduction Rules

You use the Reduction Table to add or modify the rules that govern font substitution, which you will probably never need to do. If, however, you are in Document Designer and you find that you don't like the way a font is substituted, you can change the reduction rules to substitute a different font. For information on reduction rules, see "Reduction Rules," in Section 2.

Since most of the reduction rules are created automatically, you won't be able to see which ones need to be modified; you will have to rely on informed deduction. Remember that there is a hierarchy of substitution: type style is the first attribute be be substituted, point size is next, and if a replacement font still can't be found, the font family is substituted. Figure 3-13 shows the Reduction Table.

REDUCTION TABLE

Family	Points	Bold	Ital	Device Type
*	19–23	*	*	Imagen
*	18	*	*	Imagen
Comment: 18 point is the 3rd largest point size available on the IP/III				
*	24–35	*	*	Imagen
*	24	*	*	Imagen
Comment: 24 point is the 2nd largest point size available on the IP/III				
Greek	25–99	*	*	Imagen
Greek	24	*	*	Imagen
Comment: 24 point is largest Greek on Imagen				
Cubic PS 96	*–*	*	*	
Cubic PS 96	*	*	*	Daisy
Comment: Default device is Daisy when family is Cubic PS 96				
Helvetica	*–*	*	*	Daisy
Cubic PS 96	*	*	*	Daisy
Comment: Use Cubic PS 96 in place of Helvetica on Daisy				

501A.3–13

Figure 3–13. The Reduction Table

Adding or Modifying a Reduction Rule

To add or modify a reduction rule:

1. Access the Reduction Table from the main menu.
2. Use the arrow keys to move the highlight to the reduction rule that you want to modify.
3. If you want to add a reduction rule, press **F4** (Insert) to create a blank reduction rule. If you want to modify a reduction rule, highlight the rule, press **F5** (Modify), and change the fields in the rule as required. Use **TAB** and **CODE-TAB** to move the highlight through the table.

   If necessary, use **HELP** to access the Font Families and Device Types directories.
4. Press **GO** to save the changes; press **CANCEL** to discard them.

Testing Reduction Rules

You can use the Test command to test any new reduction rules that you add.

Keep in mind that the Test command does not take into account the automatic reduction rules. Therefore, the results of the test are limited to the visible reduction rules. For information on visible reduction rules, see "Reduction Rules," in Section 2.

To use the Test command,

1. Access the Reduction Table from the main menu.
2. Press **F8** (Test) to create a blank rule. Fill in the second line of the top rule with the font you want to test. If necessary, use **HELP** to access the Font Families and Device Types directories.
3. Press **GO**. One or more reduction rules are displayed. The last rule displays the replacement font, based on the visible reduction rules. Bear in mind that the only definite way to test reduction rules is to apply fonts in Document Designer or OFIS Designer and note the replacement font in the format status line.

Generating a New Runtime Font Database

You need to generate a new runtime font database to put into effect your changes to the editable font database. Then you must copy your new runtime font database to the directory that is read by the Font Service (usually *[Sys]<Gps>*), thereby overwriting the old runtime font database with the new one.

To generate a new runtime font database, follow the steps below.

1. Access the main menu.
2. Highlight the *Generate Runtime Database* field, and press **GO**. The command is finished executing when the main menu returns to the screen.

   Generating a runtime font database can take from 30 seconds to 45 minutes, depending on its size. If the runtime font database includes raster fonts, it usually takes at least 10 minutes.

   When the runtime database is generated, you may get a status message that invalid font keys were generated. (This can happen if no similar font family is found to match your device type.) If so, you may need to create font keys or add a reduction rule. To find out what to do, enter the Font Keys directory and find the records that are flagged with bright bullets. If they are valid font keys, that is, if your printer can print them, you need to finish them. (See the section "Working With Font Keys," earlier in this section.) If they are invalid font keys, you need to delete them and create a reduction rule to insure that they won't be created again. (See the section "Working With Reduction Rules," earlier in this section.)

   One reason that invalid font keys may be generated is that there is no font key present (at all) for a device type that was being generated. There must be at least one font key, which can be derived considering any reduction rates specified, for each device type.
3. Press **FINISH** to exit the Font Tool. If the Context Manager is running, finish all applications and log out.

4. Enter the Executive, type **Deinstall Font Service** in the command line, and press **GO**.

5. Type **Copy** in the Executive command line and press **RETURN**. Use the **Copy** command to overwrite the old runtime font database with the new one.

   If you are generating the original database, the file name for the new runtime font database is *<FontDb>Font.dbs*. If you are generating a variant database, the name of the variant database is used as a prefix for the file name, for example, *<FontDb>VariantFont.dbs*.

   The default file name for the *old* runtime database (the standard Font Service) is *<Gps>Font.dbs* and *<Gps>ScalingFont.dbs* if you are using the Scaling Font Service.

   For example, if you were copying the runtime font database for the Angola variant database, the command form would look like the example shown below.

   ```
   Copy
     File from          [Sys]<FontDb>AngolaFont.dbs
     File to            [Sys]<Gps>Screenfont.dbs
     [Overwrite ok?]    Yes
     [Confirm each?]
   ```

6. Press **GO** to overwrite the old runtime font database.

7. Type **Install Font Service** and press **RETURN**.

8. In the *[Font Database]* field, enter the location of the Font Database. This is the file name you entered in the *File to* field when you overwrote the old runtime font database with the new one, for example, *[Sys]<Gps>ScreenFont.dbs*.

9. Enter **12000** in the *[Bytes of Buffer Space]* field. This value is recommended because it provides good performance while conserving memory.

10. Press **GO**.

If you are on a cluster, some workstations on the cluster may still have the old runtime font database installed on their local hard disks. If so, you should overwrite all the old runtime font databases on the cluster with the new one. It is important that all runtime font databases on a cluster be generated from the same editable font database.

The only exception to this is when one runtime font database is a superset of another one. For example, one runtime font database may include screenfonts, and another might not. It is important that information common to all runtime font databases be identical, for example, the data in width tables and translation tables.

If you want to distribute a runtime font database on floppy disks, it will probably take more than one floppy disk. To copy a runtime font database to floppy disks, use the Selective Backup command to copy it from the hard disk to the floppy disks, and then use the Restore command to copy the database from the floppy disks to the hard disk.

Converting Wheel Sets

Wheel set conversion is a partially automated procedure with prompts that tell you what to do and when to do it. If the wheel set requires a new device type, add the device type before you convert the wheel set. To add a device type, see "Working With Device Types," earlier in this section.

To convert a wheel set file,

1. In the main menu, highlight the *Convert Wheel Set File* field, and press **GO**. The *Wheel Set Conversion* form is displayed, as shown in Figure 3-14.

   For each step in the command you will see a prompt at the bottom of the form to guide you through the step. The first prompt is

   Enter Wheel File Name and press GO

   Press CANCEL to return to menu

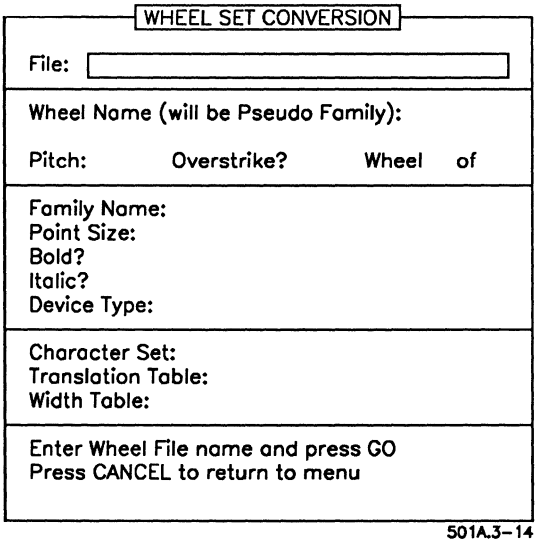


Figure 3–14. Wheel Set Conversion Form

2. Enter the wheel set file name in the *File:* field. If the file is in another volume and directory, enter the full file specification, for example, **[Sys]<Miller>Wp.whl**, or **[F0]<Grunch>Wp.whl**. Notice that the file name has the suffix **.whl**, the binary form of the file. (If you want to exit from this procedure, press **CANCEL**.)

3. Press **GO**. Four fields in the form are filled in automatically:

   Wheel Name: Displays the name of the wheel, which is also the name of the new pseudo font family.

   Pitch: Displays the pitch, if the font family is monospaced.

   Wheel _ of_ Shows how many wheels are included in the wheel set file, for example, *Wheel 1 of 2*.

   Bold? Contains an asterisk (*) since the bold attribute is applied by the printer overstriking the character. The asterisk allows this attribute to be determined by the requested font.

Italic ? This field is set to *No.* If the print wheel is an italic font, enter **yes**.

4. A new prompt is displayed at the bottom of the form:

   Press CANCEL to skip this wheel

   Press GO to create Pseudo Family

   If there is more than one wheel supported by this wheel set file, and if you don't want to convert the current one, press **CANCEL**. (The fields described in step 3 are filled in with the information belonging to the next wheel.) Otherwise, press **GO**.

5. Fill in the name of the font family in the *Family Name:* field. This can be a new font family, or you can use an existing one. If it is a new font family, you can give it any name, but it's helpful to use the name of the wheel without reference to the size or the style of the font.

   Remember that it is best to avoid creating redundant font families. For example, Elite is just 10 point Courier, so you don't need to create a new font family; just enter **Courier**.

6. Leave the *Device Type:* field blank.

7. Press **GO**. If you are creating a new font family, this message is displayed at the bottom of the form:

   Family does not exist; press GO to create or CANCEL

8. Press **GO**.

   If you press **CANCEL**, you go to the next wheel in the file. (Pressing **CANCEL** twice returns you to the main menu.)

9. A new prompt is displayed at the bottom of the form:

   Enter character set name and press GO

   Enter a name for the character set. Since this is a new character set, choose any name you like, but remember that it's helpful to use the name of the wheel.

10. Press **GO** to create a new character set.

    Another prompt is displayed at the bottom of the form:

    Enter translate table name and press GO

    Enter a name for the translation table. Once again, select any name; it should, however, be associated with the wheel name.

11. Press **GO** to create the new translation table. The *Overstrike* field is filled in automatically with *Yes* or *No*: *Yes* if the wheel uses overstrikes; *No* if it does not.

    If the wheel is for a monospaced font, go to step 13.

    If the wheel is for a proportional font, you see yet another prompt at the bottom of the form:

    Enter width table name and press GO.

    Enter a name for the width table. (Select any name.)

12. Press **GO** to create the width table.

13. A partially completed font key is displayed. Fill in the following fields:

    Family Name: Enter the name of the font family as displayed in the *Wheel Set Conversion* form.

Point Size:	Enter the point size.
Device Type:	Enter the device type. If necessary, use **HELP** to see the Device Types directory.
Alias:	Enter the name of the print wheel.

14. Press **GO**. Another partially filled in font key is displayed, which permits you to apply the same wheel to different device types.

15. If you want to add an additional font key, do so now. Press **GO** and add more font keys, as necessary. Press **CANCEL** if you don't want an additional font key.
16. If there are additional wheels in the wheel set file, repeat the above procedure from step 4. If there aren't any other wheels, the wheel set conversion is completed.

After you have finished the wheel set conversion, you need to complete the following tasks:

- Add similar font families to any new font families. Otherwise, the similar font family will be the same as the default font family.
- Generate a new runtime font database. To do so, see "Generating a New Runtime Font Database," earlier in this section.

If special characters in the wheel are created by simple overstrike commands, those commands are automatically entered into the first level translation table. However, if the wheel requires escape sequences for any characters, you have to create a second level translation table. To do so, see "Working With Translation Tables," and "Working With Character Sets," earlier in this section.

You might also want to add alternate character sets to the new font keys to access additional characters. For information on alternate character sets, see "Alternate Character Sets," in Section 2.

Exporting and Importing Data

To copy data from one editable font database to another, you need to use the Export, Export Variant, and Import commands. You can export or import data between variant databases, between original databases, from a variant to an original database, or from an original database to a variant database.

These commands, as well as two other export commands, can also be used as batch mode commands. (For more information, see "Batch Mode Commands," later in this section.)

Using the Export Commands

To export data using the Export command,

1. Enter the Font Tool and access the Font Families directory.
2. Press **F7** (Export) to add or delete daggers to the fields, as needed. Any field flagged with a dagger will *not* be exported. Records flagged with a dim bullet or a dim plus sign won't be exported either, so press **F7** to change the dim/bright status. Repeat this step for all directories in the Font Tool.

   There is a shortcut to this procedure. If you only want to export the data pertaining to certain font families or device types, enter **yes** in the *Gen?* fields in the Font Families and Device Types directories for the data you want to export, and enter **no** in the *Gen?* fields for the data you don't want to export. Then issue the Generate Runtime Database command, as explained earlier in this section. This flags all the records that you *don't* want to be exported in all the directories. At this point, if you want to export additional records, press **F7** (Export) to unflag them.
3. When all the records are flagged or unflagged as necessary, from the main menu, highlight the Export command, and press **GO**. The unflagged data is copied to port files in the *FontDb* directory. As the command is executed, the screen displays the data that is being exported so you can see if you've missed anything. For example, if you are trying to export only 2 character sets and 3 translation tables, and the screen shows that 37 raster fonts have been copied, it means that you forgot to flag a few items. In that case, flag the necessary data, and issue the Export command again.

Using the Export Variant Commands

The Export Variant command exports only the data in the variant database, that is, only the records flagged with plus signs that are bright.

To export variant data using the Export Variant command,

1. Enter the Font Tool (using a variant database), and access the Font Families directory.
2. Any field flagged with a dim plus sign will *not* be exported. Press **F7** (Export) to change the attribute from dim to bright or vice versa. Repeat this step for all directories in the Font Tool.

   There is a shortcut to this procedure. If you only want to export the data pertaining to certain font families or device types, enter **yes** in the *Gen?* fields in the Font Families and Device Types directories for the data you want to export, and enter **no** in the *Gen?* fields for the data you don't want to export. Then issue the Generate Runtime Database command, as explained earlier in this section. This changes the plus signs to dim for all the records you *don't* want to be exported. At this point, if you want to export additional records, press **F7** (Export) to make them bright.
3. When all the plus signs are dim or bright as necessary, highlight the Export Variant command on the main menu and press **GO**. The records flagged with bright plus signs are copied to port files in the *FontDb* directory. As the command is executed, the screen displays the data that is being exported so you can see if you've missed anything. For example, if you are trying to export only 2 character sets and 3 translation tables, and the screen shows that 37 raster fonts have been copied, it means that you forgot to change the dim/bright status on a few items. In that case, change the status of the flags, as needed, and issue the Export Variant command again.

Using the Import Commands

To import data using the Import command,

1. Copy the port files from the exporting Font Database to the directory of the importing Font Database. You can do this via a floppy disk or from the cluster.

   If the port files are from a variant database, they are prefixed with the name of that variant database. For example, if the variant database is named MaxFonts, the port files are named *MaxFontsFontKey.port*, *MaxFontsReduction.port*, and so forth. If the port files are from the original database, they will not have a variant prefix. This is an important point to consider when you work with multiple variant databases, because you may have several sets of port files on your system.

   Use the LCopy command to copy the port files to the directory of the importing editable font database, as shown below. Remember that your *[File prefixes from]* and *[File prefixes to]* fields may be different, depending on the locations of the Font Databases.

```
LCopy
  File list                 *.port
  [File prefix(s) from]     [!Sys]<FontDb>
  [File prefix(s) to]       [Sys]<FontDb>
  [File suffix(s)]
  [Overwrite OK?]           Yes
  [Confirm each?]
  [Continue on error?]
  [Verify copy?]            Yes
```

2. If you want to import *.port* files with a variant name prefix, you need to use the batch mode Import command, as explained in "Batch Mode Commands," later in this section.

   If you want to import *.port* files without a variant name prefix, you can use the batch mode Import command, or you can use the Import command from the Font Tool main menu. To use the Import command from the main menu, enter the Font Tool.

3. Highlight the Import command on the main menu, and press **GO**. The data from the port files is copied into the editable font database.

   The imported data is placed at the end of each directory or table. You may need to edit each directory to delete unneeded or invalid data. For example, if you imported data for a printer that you don't have, you need to delete that data.

Batch Mode Commands

Batch mode commands are often used to export or import font data. They provide a short cut because you don't have to enter the Font Tool in order to execute the comands.

The batch mode commands are also used to distribute a custom version of an editable font database with distribution diskettes. For example, these commands are used during part of the initial GPS installation procedure.

The Font Tool batch mode commands are described below.

Create	As explained in "Creating a Variant Database" earlier in this section, this command is used to create a variant database. It is also used when you set up an editable font database directory. This command creates empty files that port files can be imported into. When you create a variant database, it creates empty files in the variant database.
Exit	Used only in submit files to verify that the Font Tool is already installed. The Exit command opens the editable font database and then closes it. If the editable font database has not been installed, however, it can't be opened, and the submit file cannot be completed.

Export	Functions the same as the Export command described earlier in this section. (See the section "Exporting and Importing Data.") When used as a batch mode command, you can specify which variant database to export data from by entering the name of the variant database in the *[Variant database]* field of the Font Tool command form. The example below shows how to export data from the variant database *MaxFonts*. If you leave the *[Variant database]* field blank, you export data from the original database.

Font Tool

[Batch mode commands] Export

[Variant database] MaxFonts

ExportAll	Exports *all* data in the editable font database, whether or not it is flagged with a dagger, dim bullet, or dim plus sign. If you specify a variant database in the *[Variant database]* field, it exports all records in that database, including original records.
ExportAll-Variant	Exports all variant data in a variant database, whether or not the plus sign is dim or bright. Any record without a variant status flag is not exported. As with the other export commands, you can specify which variant database to export from in the *[Variant database]* field.
ExportVariant	Functions the same as the Export Variant command described earlier in this section. This command exports only referenced variant records, in other words, records flagged with bright plus signs. (See "Exporting and Importing Data," earlier in this section.)

Generate

Generates a new runtime font database. You can use the *[Variant database]* field to specify which variant database to generate a runtime font database from. For example, to generate a runtime font database from the *MaxFonts* variant database, complete the command form as shown below. If the *[Variant database]* field is left blank, the runtime font database is generated from the original database.

```
Font Tool
    [Batch mode commands]     Generate
    [Variant database]        MaxFonts
```

The new runtime font database is always named *Font.dbs*, unless it is generated from a variant database. In that case, it has the name of the variant database as a prefix, for example, *VariantFont.dbs*. In any case, the runtime font database file is placed in the current directory, which is usually *FontDb*.

Import

Imports data into the editable font database. Once again, you can specify which database to import into by using the *[Variant database]* field. If the *[Variant database]* field is left blank, the port files are copied into the original database.

You can also use this command to specify the port files to be imported. For example, if you want to import port files from a specific variant database, you would enter the name of the variant database in the *[Batch mode commands]* field, along with the import command. This is useful when you work with multiple variant databases and you only want to copy port files with a specific prefix. The example below shows how to import the port files from the *Angola* variant database into the *MaxFonts* variant database. Note that the **Import** command is followed by a colon (:), then the name of the variant database to be imported.

```
Font Tool
    [Batch mode commands]     Import:Angola
    [Variant database]        MaxFonts
```

Using Batch Mode Commands

The batch mode commands are executed through the Executive, as follows:

1. Path to *[Sys]<FontDb>*.
2. Type **Font Tool** in the command line.
3. Press **RETURN**.
4. In the *[Batch mode commands]* field, enter the name of the command. For example, if you want to export the entire editable font database, enter ExportAll, as shown below.

```
Font Tool
   [Batch mode commands]    ExportAll
   [Variant database]
```

5. If you want the command to apply to a variant database, enter the name of the variant database in the *[Variant database]* field. If this field is left blank, the command will apply to the original database. The example below shows how to export all the data from the *MaxFonts* variant database.

```
Font Tool
   [Batch mode commands]    ExportAll
   [Variant database]       MaxFonts
```

6. Press **GO**.

Sometimes you can use two batch mode commands at once. For example, you can make a copy of a variant database by using the Create command in conjunction with the Import command. In the example below, a new variant database, *ZoeFonts*, is created, and the port files from another variant database, *MaxFonts*, are immediately copied into it. This is useful if you are using multiple variant databases.

```
Font Tool
   [Batch mode commands]    Create Import:MaxFonts
   [Variant database]       ZoeFonts
```

Section 4
Using the Scaling Font Service

This section describes tools used to modify the Font Database. The Scaling Font Service is an optional product used in place of the Font Service. In addition to the capabilities of the Font Service, the Scaling Font Service scales fonts on-the-fly for use or for display (by applications such as Document Designer and OFIS Designer) or to be downloaded to printers supported by the HP LaserJet device driver.

Overview

The Scaling Font Service is used instead of the Font Service and the Screen Font Database. Its package consists of outlines for the 35 fonts resident in LaserWriter II printers and a font database that supports the display of bit-maps to applications such as OFIS Designer and Document Designer, to GPS device drivers, and to applications linked with the CCGI+ library.

These bit-maps are available at any size, in any resolution, and for any orientation supported by the applications listed above. To deliver this flexibility and selection, the Scaling Font Service creates bit-maps on-the-fly from the 35 font outlines. For example, the Scaling Font Service provides characters in up to

- *38 point* type when using OFIS Designer 2.0 and Document Designer 2.0, 2.1, and 2.2
- *76 point* type when using OFIS Designer 2.1 and Document Designer 2.3
- *64 pixels* in height when using applications built with the CCGI+ library

Components of the Scaling Font Service

The Scaling Font Service software installation consists of the following components:

- Scaling Font Service

  Provides the same data to the GPS device drivers and applications as is provided by the Font Service and the Screen Font Database.

- Font Database Port Files

  Which are loaded into an editable font database. Port files include information about the scaling font outlines only. They do not include data for fonts distributed with GPS. Instead, these port files are imported into a font database *along with* data from GPS.

- 37 Master International Font Outline Files

  Includes 35 font outline files which correspond to the fonts listed above, including all the characters in the PostScript character sets, and two additional font outline files (Dutch Roman and Swiss), which contain the standard Bitstream character set including many foreign and special symbols not included in the PostScript character sets.

- 37 Nationalized Font Outline Files

  Includes 33 font outline files which correspond to the fonts listed above (except Greek and ITC Zapf Dingbats), and which contain only those characters required for the CTOS character set. It also includes four miscellaneous font outline files for supplementary characters drawn from the Swiss, Dutch Roman, Symbol, and Helvetica fonts.

- Font Outline Nationalization Tool

  Automatically builds a nationalized outlines file for a character set that you define.

- Export Raster Font Utility

  Retrieves scaled fonts from the Scaling Font Service and generates *.port* files so that these scaled fonts can be imported to the font database. As a result, you can minimize font retrieval time by

scaling commonly used fonts in advance. Before proceeding, consider the fact that doing so increases the size of the runtime font database. As a result, it will occupy more disk space.

Obtaining Prescaled Fonts

If you install the editable font database and its tools, you can prescale fonts for use with the Scaling Font Service. As a result, prescaled fonts can be retrieved extremely quickly. You use the Export Raster Fonts utility to prescale fonts.

Export Raster Fonts
[Families] _____
[Sizes (12)] _____
[Device type (Video120)] _____
[Prefix for uniqueness] _____

Export Raster Fonts obtains scaled raster fonts from the Scaling Font Service for each font family and for each font size you list. If you do not specify a font size, Export Raster Fonts retrieves fonts in the 12 point size only. If you do not specify a font family, Export Raster Fonts obtains fonts for each family for which scaling fonts are available. Finally, Export Raster Fonts obtain fonts for each style which is applicable to each font family.

In most cases, Export Raster Fonts obtains the following four styles: Roman, Italic, Bold, and Bold Italic. (This does not apply to Greek, ITC Zapf Chancery, and ITC Zapf Dingbats, which are available in one style only.)

The following procedure summarizes the commands you use to add prescaled fonts to your font database. It creates 36 scaled fonts (the product of 3 families x 3 sizes x 4 styles). All these commands are issued in the Executive. For more information on using the Executive, see the *CTOS Executive User's Guide.*

Install Scaling Font Service

Path
- [Volume] _____
- [Directory] FontDB

Export Raster Fonts
- [Families] Courier Times Helvetica
- [Sizes (12)] 10 12 14

Font Tool
- [Batch mode commands] Import:Scaled Generate
- [Variant database] _____

Deinstall Font Service

LCopy
- File list Font.dbs
- [File prefix(s) from] _____
- [File prefix(s) to] [Sys]<Gps>Scaling
- [File suffix(s)] _____
- [Overwrite OK?] Yes

Install Scaling Font Service

Each time you create a scaled font and add it to the font database, you increase the size of the font database. If it becomes necessary to remove some scaled fonts from the database, for example to save disk space, you can use the Font Tool to manually remove the corresponding font keys. A bullet status flag indicates the scaled font keys. You may also wish to remove the corresponding records from the Raster Fonts directory. To do so, use the Delete command in the Executive to remove the font files themselves.

If you think you might want to generate a database with fewer fonts, you can save yourself some effort by storing the scaled fonts in a variant database. To do so, follow the steps above, but modify the Font Tool command so that it contains a name for a variant database (see "Creating a Variant Database" in Section 3 for details):

```
Font Tool
   [Batch mode commands]    Import:Scaled Generate
   [Variant database]       MyScaled
```

Later, to start over, simply delete the variant database called *MyScaled*.

```
Delete
   File list                MyScaled*
   [Confirm each?]
```

Successive executions of the Export Raster Fonts command overwrite the *.port* files. Therefore, it you want to export different selections of sizes for different selections of families, you must import each time.

Note: *In this example, quotes are used to indicate that New Century Schoolbook is one font and not three. Alternatively, to generate several fonts, you can list their names in a file and then use an @ file to invoke that list.*

```
Export Raster Fonts
   [Families]               Courier Times Helvetica
   [Sizes (12)]             10 12 14

Font Tool
   [Batch mode commands]    Import:Scaled Generate
   [Variant database]

Export Raster Fonts
   [Families]               'New Century Schoolbook' Palatino
   [Sizes (12)]             12

Font Tool
   [Batch mode commands]    Import:Scaled Generate
   [Variant database]
```

Fonts prescaled by Export Raster Fonts are used by the Scaling Font Service whenever the form of the application request allows the prescaled font to be used and when the size of the requested font matches the size at which the font was prescaled. For instance, OFIS Designer and Document Designer use presecaled fonts when they are available. Export Raster Fonts has two additional parameters, which are rarely, if ever, used. They are *[Device type]* and *[Prefix for Uniqueness].* You can specify a device type other than Video120 in the *[Device type]* parameter. OFIS Designer and Document Designer currently use only 120 dots per inch (dpi) fonts for display, which is defined by the default Video120 font device type. Other applications might use fonts of other resolutions, as defined by other font device types.

When you specify a different device type, you should also specify a short *[Prefix for Uniqueness]* to insure that the names of the raster font files for the device type you choose do not collide with the names of the raster font files generated for other device types. The names of exported raster font files are created by concatenating the prefix, the name of the outline file, an underscore, the point size of the font, an M if the font is monospaced, and *.font.*

There are other situations in which these names may not be sufficiently unique to distinguish scaled fonts which are different. For example, the same outline could be used in two different font keys for a device type with different character set definitions. In these circumstances, it may be necessary to use different prefixes when generating prescaled fonts for different families. To avoid all possible collision problems for prescaled fonts generated with the standard scaling font database, use the *Prefix for Uniqueness* field *only* when you prescale the Greek and ITC Zapf Dingbats fonts.

CAUTIONS

You cannot prescale Video120 fonts to sizes greater than 36 points because the bitmap font file format does not support sizes larger than 36 points.

The Scaling Font Service must be installed locally to execute Export Raster Fonts. If not, Error 45 occurs.

Export Raster Fonts does not create font keys suitable for use with the standard Font Service.

Use of fonts prescaled by the Scaling Font Service is restricted to use on those clusters for which the Scaling Font Service has been purchased.

Configuring the Scaling Font Service

You use the Font Tool delivered with GPS to build the font database used with the Scaling Font Service. The Scaling Font Service will scale a font if the font key names a device type for which *Rasters?* has been set to **yes** and the *Scale by Points?* field of the Font Key is set to **yes**. *Scale by Points?* should be set to **yes** in each Font Key for a scaling font.

You can leave the *Raster Font* field blank. If the Font Tool displays a name in this field, it is the name of the previously scaled font file. You can use this previously scaled font file for faster access provided the request was for the exact same point size. Keep in mind that you must specify a width table for proportionally spaced scaling font keys.

Most applications that obtain scaled fonts from the Scaling Font Service rely on the device type record, referenced in each font key, to define the resolution of the device. The resolution of the device, together with the point size of the requesting Font Key, determines the height and width, in pixels, of the scaled bit-maps.

The Export Raster Fonts utility creates Font Keys, in a *.port* file, for each of the specific sizes of the font that is being rasterized. The utility copies the font key for the unscaled font and not only puts the font in the specified size but also inserts the name of the scaled font file next to *Raster Font* and adjusts *Width Factor* for the new size. For example, if *Width Factor* for a 1 point font key is 1/1. then the *Width Factor* for a 12 point font key is 12/1.

The character set named in the Font Key requires both a *From CTOS* (level 1) translation table and a *To Printer* (level 2) translation table. The level 2 translation table is generally stored in order of the PostScript character set. One reason for this is that it makes it easier to create width tables for the scaling fonts by copying width tables for corresponding PostScript fonts. For example, the level 2 translation table maps an 8-bit characer code to the Bitstream character ID, which can be either an 8-bit number (2 hexadecimal digits) or a 16-bit number (four hexadecimal digits).

This character ID is useful since for some fonts there are more characters than there are unique 8-bit codes. For this reason, not all PostScript characters are assigned codes. To use PostScript characters without codes, you must do two things. First, you assign an 8-bit code to the character and then, in the level 2 translation table, you store the Bitstream character ID in the position of that 8-bit code.

This is similar to the way that characters are identified in the PostScript and Imagen fonts. The next section lists Bitstream character IDs and shows you how to map each character for access on PostScript printers as well.

The level 1 translation table maps the characters in the nationalized CTOS character set to the position in the level 2 translation table that contains the Bitstream character ID. As stated earlier, this is usually the character code that PostScript assigns to the character. The level 1 translation table can also contain escape codes to coerce the width of the character to that of another character (the 10 escape code) or to shift it vertically (the 05 and 06 escape codes).

The translation tables named in the character set are used only to map the character assignments, to shift characters, or to coerce character widths when scaling fonts on-the-fly. These operations are already performed when a font is prescaled by the Export Raster Fonts utility. Consequently, the font files created by Export Raster Fonts are already reordered in the CTOS character set order.

The Font Tool assumes that the font files are already ordered in the character set or whenever *Scale by Size*? is set to *Yes* in the Font Key Detail form. For example, the Font Tool assumes that the character sets are specified for the purpose of on-the-fly scaling, not for the purpose of reordering the font file. The character set is used by the Font Tool to shuffle the ordering of fonts only when *Scale by Size?* is set to *No*.

Font Nationalization

The following material describes how to customize or nationalize the character mapping of the Latin-based fonts in the Scaling Font Service. (The Greek and ITC Zapf Dingbats fonts are not discussed since they are direct mappings of the PostScript character assignments and should not require nationalization.) It describes the mappings for both scaling screen fonts and PostScript fonts since a match between the screen display and the PostScript printed result is desired.

The scaling fonts outline files contain a number of useful symbols for which there are not corresponding CTOS character codes. Reassignment of character codes to glyphs present in the raster font files is easily performed by modifying the translation table maintained by the Font Tool, which is used to map the CTOS character set to glyphs in each font. Of course, printer translation tables should also be changed in this case so that corresponding characters in printer fonts are used.

Note: *To perform these customizations, create a variant database.*

Character Codes

The character codes generally available for nationalization or redifinition are those in the ranges A0-B5, 83-87, B7-BC, BE, and BF. You can customize the system font in these ranges to affect corresponding changes in non-wysiwyg Document Designer or OFIS Designer. Since the CTOS character codes in the range C0-FF have no equivalent characters on most printers, they too can also be customized. However, it is impossible to change the system fonts in this range since many software packages use these codes for on-screen rulings.

Table 4-1 contains information on characters in the Scaling Font database. Refer to the following descriptions to understand the contents of the columns in Table 4-1.

Description

Description of the character. If the first character in the description is capitalized, then the description is for an uppercase character. Likewise, if the first character is not capitalized, the character is either lowercase or caseless.

Class

Indicates one of three font classes. The three font classes have the precedence enumerated below. In other words, translations using a higher class override translations using a lower class. For example, if you change the translation of a character that uses a character in class Text, to use a character in class Sym (lower prioroty), the translation in CtToBitstreamStandard must be removed at the same time you add the translation to CtToBitstreamSymbol. The three font classes are:

- *Text* indicates that the character is in the standard PostScript character set.

  The Bitstream character set is called BSStandard and the translation tables are CtToBitstreamStandard (level 1) and BitstreamStandard (level 2). The PostScript character set is called PostScriptN and the translation tables are CtToPostScriptN (level 1) and PostScriptNToNatl (level 2).

- *Sym* indicates that the character is drawn from the PostScript symbol font.

  The Bitstream character set is called BSSymbol and the translation tables are CtToBitstreamSymbol (level 1) and BitstreamSymbol (level 2). The PostScript character set is called PostScriptS and the translation table is CtToPostScriptS (level 1; there is no level 2 translation table).

- *Misc* indicates that the character is drawn from the Bitstream Swiss or Dutch Roman fonts.

  The Bitstream character set is called BSMisc and the translation tables are CtToBitstreamMisc (level 1) and BitstreamMisc (level 2). These characters cannot be printed on PostScript printers.

CTOS

Contains the BTOS or CTOS character code that maps to the described character in the font database distributed by Unisys. Of course, many characters do not have a corresponding CTOS character code; that's why users nationalize the CTOS character code set. Dashes denote these characters. The CTOS character codes range from 21 to 7E and include the printable ASCII characters known worldwide, including all the unaccented characters in the Latin-based alphabet.

We omitted most of these from the table because there are so few conditions that require them to be modified. However, we did include several of the more ambiguous characters that might require different mappings in certain environments. For example, is ASCII 2D a hyphen, a minus sign, or a dash? In the days of typewriters and teletype machines, the same character could be used in each case. Today, however, the characters look quite different from each other in the printed book.

Where an alternative mapping of a standard CTOS character could be argued for, we have included the mapping in the subsequent row. In these cases, the CTOS code is displayed in parentheses. Also, if the mapping of the PostScript character differs from that in the font database in releases of CTOS GPS 2.3 or earlier, the old mapping is indicated by an asterisk in the CTOS column.

Bitstream Levels 1 and 2

Level 1 contains the translation of the character in the first level translation table for the scaling font. It is the character code that the Scaling Font Service sends to the printer when the font is downloaded to an HP LaserJet-compatible printer. Bitstream Level 2 contains the corresponding Bitstream character ID which is stored in the second level translation table, in the slot indicated by the first level translation.

In general, follow these steps when you need to assign a different character to a character code:

1. Modify the first level translation table, according to the class, to indicate the Bitstream Level 1 character code that is shown for the desired character.
2. If the Bitstream Level 1 column shows a dash for a character that you need, you must modify both the level 1 and level 2 translation tables.
3. Select an 8–bit character code at level 1 that you won't be using for any other character in that class, or any class sharing the same character set.
4. Put the Bitstream character ID shown in the Bitstream Level 2 column in that position of the second level translation table.

PostScript Levels 1 and 2

Level 1 contains the translation of the character in the first level translation table for the font. It is the character code that the Scaling Font Service sends to the PostScript printer. If this character does not have a preassigned character code in PostScript, PostScript Level 2 contains a 16-bit (4-digit hexadecimal) code to identify the character to the GPS PostScript device driver.

The device driver instructs the PostScript printer to assign the character to the character code indicated in the PostScript Level 1 column. If a dash appears in PostScript Level 2, the 2nd level translation is nil because the character code is preassigned. But if a dash appears in both columns, it is a character for display only, or a character which cannot be printed on PostScript printers. A ? following a hexademical translation in PostScript Level 1 indicates that this is not a preassigned PostScript character code. As with the scaling fonts, to assign a different character to a character code:

1. Modify the first level translation table (according to the class) to indicate the PostScript Level 1 character code that is shown for the desired character.
2. If the PostScript Level 1 column shows only a ? for a character that you need, you must modify both the level 1 and level 2 translation tables.
3. To do so, select an 8-bit character code at level 1 that you won't be using for any other character in that class, or class sharing the same character set.
4. Put the PostScript character ID shown in the PostScript Level 2 column in that position of the second level translation table.

PostScript character codes must fall into the one of the two ranges, 21 to 7E and A1 to FE. Also, character codes used to access downloaded characters on HP LaserJet printers must reside in the same ranges. The HP LaserJet device driver uses the same translation tables as the scaling screen fonts, but characters whose first level translation is not in the required ranges are not downloaded. Any code other than 00 can be used to display characters on the screen; therefore, we use codes below 20 hex for characters that are displayed only, such as the visible outline paragraph. We assigned the same Bitstream character code as PostScript code whenever a PostScript code is assigned. This simplifies the maintenance of the width tables.

Table 4–1. Characters Available for Nationalization

Description	Class	CTOS	Bitstream		PostScript	
			Level 1	Level 2	Level 1	Level 2
up arrow	Sym	01	AD	01CD	AD	–
3/4	Text	02	C0?	BE	C0?	0200
cent	Text	03	A2	84	A2	–
filled diamond	Sym	04	A8	024D	A8	–
unfilled diamond	Sym	05	E0	17D5	E0	–
1/2	Text	06	C9?	BC	C9?	0600
vis. tab	Sym	08	DC	12E8	–	–
vis. backspace	Sym	09	DE	12E1	–	–
vis. carrige return	Sym	0A	BF	476B	–	–
down arrow	Sym	0B	AF	01CA	AF	–
vis. form feed	Sym	0C	DF	12D9	–	–
left arrow	Sym	0E	AC	01CB	AC	–
double dagger	Text	0F	B3	8F	B3	–
1/4	Text	10	CC?	BA	CC?	1000
dagger	Text	11	B2	8E	B2	–
right arrow	Sym	12	AE	01CC	AE	–
en dash	Text	13	B1	91	B1	–
non–break space	Text	14	20	20	20	–
divide	Sym	15	B8	01C3	B8	–
divide	Text	(15)	0115	01C3	?	1500
vertical bar	Sym	16	BD	17E9	BD	–
section symbol	Text	17	A7	90	A7	–
not equal	Sym	18	B9	01C6	B9	–
approx. equal	Sym	19	BB	01C7	BB	–
paragraph symbol	Text	1A	B6	01BA	B6	–
bullet	Text	1B	B7	023F	B7	–
not	Text	1C	BE?	01E6	BE?	1C00
not	Sym	(1C)*	D8	01E6	D8	–
less or equal	Sym	1D	A3	01C9	A3	–
plus or minus	Sym	1E	B1	01C4	B1	–
plus or minus	Text	(1E)	011E	01C4	?	1E00
greater or equal	Sym	1F	B3	01C8	B3	–
apostrophe	Text	27	27	27	27	–
single quote	Text	(27)	A9	019A	A9	–
acute accent	Text	(27)	C2	A3	C2	–
hyphen	Text	2D	2D	2D	2D	–
en dash	Text	(2D)*	B1	91	B1	–
minus	Text	(2D)	0113	01C1	?	1300
slash	Text	2F	2F	2F	2F	–

continued

Table 4–1. Characters Available for Nationalization (cont.)

Description	Class	CTOS	Bitstream		PostScript	
			Level 1	Level 2	Level 1	Level 2
fraction bar	Text	(2F)	A4	87	A4	–
ascii circumflex	Text	5E	5E	5E	5E	–
circumflex accent	Text	(5E)	C3	A7	C3	–
open single quote	Text	60	60	89	60	–
grave accent	Text	(60)	C1	A5	C1	–
vertical bar	Text	7C	7C	7C	7C	–
broken vertical bar	Text	(7C)	0116	01B3	?	1600
tilde	Text	7E	7E	1288	7C	–
tilde accent	Text	(7E)	C4	AB	C4	–
non–breaking hyphen	Text	80	2D	2D	2D	–
optional hyphen	Text	81	2D	2D	2D	–
optional space	Text	82	20	20	20	–
leader dot	Text	88	2E105F	2E	2E105F	–
vis. para mark	Text	89	B6	01BA	–	–
vis. heading para	Text	8A	A7	90	–	–
vis. TOC para	Misc	8B	0104	0237	–	–
vis. outline para	Misc	8C	0105	0238	–	–
vis. null	Sym	8D	C6	1389	–	–
vis. space	Text	8E	B41020	88	–	–
vis. non–break space	Misc	8F	06263D1020	1178	–	–
vis. optional space	Misc	90	5F1020	06DC	–	–
vis. non–break hyphen	Text	91	060A1C102D	01E6	–	–
vis. optional hyphen	Text	92	0627C4102D	AB	–	–
A ring	Text	A0	B0?	93	B0?	41CA
a ring	Text	A1	D1?	97	D1?	61CA
A diaeresis	Text	A2	D2?	0189	D2?	41C8
a diaeresis	Text	A3	D3?	018A	D3?	61C8
O diaeresis	Text	A4	D4?	014D	D4?	4FC8
o diaeresis	Text	A5	D5?	014C	D5?	6FC8
O slash	Text	A6	E9	95	E9	–
o slash	Text	A7	F9	99	F9	–
U diaeresis	Text	A8	D8?	0157	D8?	55C8
u diaeresis	Text	A9	D9?	0156	D9?	75C8
c cedilla	Text	AA	DA?	B8	DA?	63CB
e circumflex	Text	AB	DB?	0182	DB?	65C3
e acute	Text	AC	DC?	0186	DC?	65C2
e grave	Text	AD	DD?	0184	DD?	65C1
AE ligature	Text	AE	E1	94	E1	–
ae ligature	Text	AF	F1	98	F1	–

Table 4–1. Characters Available for Nationalization (cont.)

Description	Class	CTOS	Bitstream		PostScript	
			Level 1	Level 2	Level 1	Level 2
German double s	Text	B0	FB	9B	FB	–
pound sterling	Text	B1	A3	83	A3	–
ring accent	Text	B2	CA	B5	CA	–
degree	Text	(B2)	0101	023C	?	0100
degree	Sym	(B2)	B0	023C	B0	–
copyright	Text	B3	FC?	0234	FC?	0800
sans copyright	Sym	(B3)*	E3	2354	E3	–
serif copyright	Sym	(B3)	D3	21A6	D3	–
registered	Text	B4	FD?	0235	FD?	0900
sans registered	Sym	(B4)*	E2	2352	E2	–
serif registered	Sym	(B4)	D2	21A4	D2	–
trademark	Text	B5	FE?	0236	FE?	0A00
sans trademark	Sym	(B5)*	E4	2357	E4	–
serif trademark	Sym	(B5)	D4	21A9	D4	–
inverted !	Text	–	A1	A2	A1	–
yen	Text	–	A5	01AB	A5	–
florin	Text	–	A6	85	A6	–
currency	Text	–	A8	01B4	A8	–
open double quote	Text	–	AA	8A	AA	–
double left guillemot	Text	–	AB	9F	AB	–
left guillemot	Text	–	AC	9D	AC	–
right guillemot	Text	–	AD	9E	AD	–
fi ligature	Text	–	AE	81	AE	–
fl ligature	Text	–	AF	82	AF	–
centered dot	Text	–	B4	88	B4	–
baseline single quote	Text	–	B8	8C	B8	–
baseline double quote	Text	–	B9	8D	B9	–
close double quote	Text	–	BA	8B	BA	–
double right guillemot	Text	–	BB	A0	BB	–
three leader dots	Text	–	BC	01A7	BC	–
per mil	Text	–	BD	86	BD	–
inverted ?	Text	–	BF	A1	BF	–
macron accent	Text	–	C5	B1	C5	–
breve accent	Text	–	C6	B3	C6	–
dot accent	Text	–	C7	ED	C7	–
diaeresis accent	Text	–	C8	A9	C8	–
cedilla accent	Text	–	CB	AF	CB	–
double acute accent	Text	–	CD	EF	CD	–
ogonek accent	Text	–	CE	F6	CE	–

Table 4–1. Characters Available for Nationalization (cont.)

Description	Class	CTOS	Bitstream		PostScript	
			Level 1	Level 2	Level 1	Level 2
caron accent	Text	–	CF	AD	CF	–
em dash	Text	–	D0	01A2	D0	–
feminine ordinal	Text	–	E3	0192	E3	–
masculine ordinal	Text	–	EB	0191	EB	–
L slash	Text	–	E8	E2	E8	–
OE ligature	Text	–	EA	96	EA	–
dotless i	Text	–	F5	9C	F5	–
slash	Text	–	F8	E6	F8	–
oe ligature	Text	–	FA	9A	FA	–
A acute	Text	–	?	018F	?	41C2
A circumflex	Text	–	?	018B	?	41C3
A grave	Text	–	?	018D	?	41C1
A tilde	Text	–	?	0187	?	41C4
C cedilla	Text	–	?	B7	?	43CB
E acute	Text	–	?	0185	?	45C2
E circumflex	Text	–	?	0181	?	45C3
E diaeresis	Text	–	?	017F	?	45C8
E grave	Text	–	?	0183	?	45C1
I acute	Text	–	?	0179	?	49C2
I circumflex	Text	–	?	0175	?	49C3
I diaeresis	Text	–	?	0173	?	49C8
I grave	Text	–	?	0177	?	49C1
N tilde	Text	–	?	0143	?	4EC4
O acute	Text	–	?	0147	?	4FC2
O circumflex	Text	–	?	014B	?	4FC3
O grave	Text	–	?	0149	?	4FC1
O tilde	Text	–	?	014F	?	4FC4
S caron	Text	–	?	029B	?	53CF
U acute	Text	–	?	0151	?	55C2
U circumflex	Text	–	?	0155	?	55C3
U grave	Text	–	?	0153	?	55C1
Y acute	Text	–	?	015F	?	59C2
Y diaeresis	Text	–	?	015D	?	59C8
Z caron	Text	–	?	0264	?	5ACF
a acute	Text	–	?	0190	?	61C2
a circumflex	Text	–	?	018C	?	61C3
a grave	Text	–	?	018E	?	61C1
a tilde	Text	–	?	0188	?	61C4
e diaeresis	Text	–	?	0180	?	65C8

Table 4–1. Characters Available for Nationalization (cont.)

Description	Class	CTOS	Bitstream		PostScript	
			Level 1	Level 2	Level 1	Level 2
i acute	Text	–	?	017A	?	69C2
i circumflex	Text	–	?	0176	?	69C3
i diaeresis	Text	–	?	0174	?	69C8
i grave	Text	–	?	0178	?	69C1
n tilde	Text	–	?	0142	?	6EC4
o acute	Text	–	?	0146	?	6FC2
o circumflex	Text	–	?	014A	?	6FC3
o grave	Text	–	?	0148	?	6FC1
o tilde	Text	–	?	014E	?	6FC4
caron	Text	–	?	029A	?	73CF
u acute	Text	–	?	0150	?	75C2
u circumflex	Text	–	?	0154	?	75C3
u grave	Text	–	?	0152	?	75C1
y acute	Text	–	?	015E	?	79C2
y diaeresis	Text	–	?	015C	?	79C8
z caron	Text	–	?	0263	?	7ACF
1 superior	Text	–	?	C2	?	0300
2 superior	Text	–	?	C3	?	0400
3 superior	Text	–	?	C4	?	0500
mu	Text	–	?	0216	?	0B00
Eth	Text	–	?	E1	?	0C00
eth	Text	–	?	01A8	?	0D00
Thorn	Text	–	?	01AA	?	0E00
thorn	Text	–	?	01A9	?	0F00
multiply	Text	–	?	01C2	?	1400

Modifying Width Tables After Nationalization

If, during the course of nationalization, you must modify the 2nd level translation table BitstreamStandard or PostScriptNToNatl, usually to access accented characters other than those in the CTOS character set, you must modify all of the affected width tables. The widths of these unaccented characters are not stored in the width tables because there isn't enough space to store the width of every character.

Fortunately, the width of accented characters is invariably the same as the width of unaccented characters. The Font Tool command Propagate actually propagates the width of unaccented characters to positions occupied by accented characters. The Propagate command requires a translation table as its parameter. The peculiar scheme used to indicate each of the unassigned PostScript characters was designed to exploit the Propagate command for updating width tables. Notice that for accented characters, the first two hexadecimal digits of the PostScriptNToNatl translation is the ASCII code of the corresponding unaccented character.

After modifying PostScriptNToNatl:

1. Access the Width Table directory.
2. Press **F5** (Modify) for each Width Table that shows the PostScriptN character set.
3. Press **F9** (Prop).
4. Type **PostScriptNToNatl**
5. Press **GO** twice (once to propagate and once more to store the updated width table).
6. On subsequent width tables:
   a. Press **F9** (Prop).
   b. Press **F1** (Redo). PostScriptToNatl is automatically entered in the field.
   c. Press **GO**.

If you modify PostScriptNToNatl, you probably need to modify the BitstreamStandard translation table too. Use the same translations at Level 1 for both Bitstream and PostScript for the accented characters, so that the same positions in the width tables for the scaling fonts and PostScript fonts hold the widths. Then go to the Width Table directory, and propagate widths for all width tables which show the BSStandard character set. Use the PostScriptNToNatl table again to propagate widths. (The BitstreamStandard translation table uses Bitstream character IDs that are not understood by the Propagate command.)

You can use the Propagate command over and over again, for as many times as you modify the PostScriptNToNatl table. However, beware of using PostScript positions which are used for preassigned characters unless you know that you will *never* use them. Once you propagate widths, the width of the preassigned character is permanently lost (unless you can look up and rekey each one manually). The safe PostScript positions to reassign (they are not preassigned) and for which width tables do not contain unrecoverable data are B0, B5, BE, C0, C9, CC, D1-E0, E2, E4-E7, EC-EF, F0, F2-F4, F6, F7, FC-FE.

Not all of the unassigned PostScript characters are accented characters. We stored the widths of these other characters in locations below 20 hex, which are not used when printing to PostScript printers. To use these characters, you must assign them positions in the legal range of PostScript characters, just as you do for accented characters. When you propagate widths, the widths of these characters are moved into the asssigned positions. (The first two bytes of the second level PostScript translation of these characters is the position of the width table reserved to hold the width of the character permanently.)

You do not need to modify BitstreamStandard to access these characters for display. BitstreamStandard already contains the Bitstream character IDs in the same positions below hex 20, since codes below hex 20 are acceptable for output to screens. If you need to download these characters to an HP LaserJet-compatible printer, you must assign the same code in the legal range that you used for PostScript and propagate the widths for the corresponding width tables.

Nationalize Scaling Fonts Utility

After changing the translation tables for the Scaling Font Service, you must execute the Nationalize Scaling Fonts command to regenerate the nationalized font outline files, which have names matching the pattern. To create the outline files, extract the characters referenced in the font database from the corresponding complete font outline files. The command form for Nationalize Scaling Fonts is:

```
Nationalize Scaling Fonts
  [Families]                    _____________________________
  [Device type (Video120)]      _____________________________
```

After modifying the translation tables in the Font Tool, you must:

1. Generate a new runtime font database.
2. Deinstall the font service.
3. Delete the old nationalized font outline files from the directory accessed by the Scaling Font Service.
4. Install the Scaling Font Service, using the new runtime font database.
5. Path to the directory which contains the complete font outline files.
6. Execute the Nationalize Scaling Fonts command. This command obtains the modified translation tables from the Scaling Font Service to determine which characters to extract from each outline file.
7. Copy the resulting nationalized outline files to the directory which the Scaling Font Service is accessing, if it is different from the directory in which the new outline files are created.

WARNING

Do not run any application, such as OFIS Designer or Document Designer, after installing a font service with new scaling font translation tables until the Nationalize Scaling Fonts command has been executed and the new nationalized outline files are put in place. To guard against this, delete the obsolete files *before* reinstalling the Scaling Font Service.

The Nationalize Scaling Fonts command normally requires about 30 minutes. The Nationalize Scaling Fonts command fails to nationalize any outline font file which does not contain all the characters specified in the translation tables. The Nationalize Scaling Fonts command proceeds to the next font when nationalization of an outline font file fails for any reason.

You can test changes to translation tables one or more times before running the Nationalize Scaling Fonts command to completion. The Nationalize Scaling Fonts command form takes, as an optional parameter, the names of families to nationalize. Thus, by default, the Nationalize Scaling Fonts command nationalizes all font families in the font database for which scaling fonts are specified.

If you nationalize just one family (Courier, for example) the command executes much faster, and you may then test the translation table changes using Document Designer and the Courier font. When you have finished testing, execute Nationalize Scaling Fonts with no parameters.

WARNING

Some nationalized fonts are created by combining characters required by different families. Therefore, executing Nationalize Scaling Fonts for each font family individually does not give the same results as Nationalize Scaling Fonts for all families at once. You must execute Nationalize Scaling Fonts with no parameters before using the nationalized font outlines in a production environment.

Nationalize Scaling Fonts has one other optional parameter which rarely needs to be specified. The device type can be specified if outlines for device types other than Video120 must be nationalized. This won't be necessary if the other font device type uses the same outlines and character sets (or a subset thereof) as Video120.

The HpScale font device type meets this criterion. However, if another font device type uses different character sets or outlines, you may need to nationalize those fonts as well. Nationalized outline files for other device types which are nationalized using different character sets must have different names than those nationalized for the Video120 device type, in order to avoid collisions.

Nationalization of Utilities

If the file *[sys]<sys>FontNatMsg.bin* is present on your system, the utilities Export Raster Fonts and Nationalize Scaling Fonts display messages in the native language, as defined in that message file. If the message file is inaccessible, messages are displayed in English.

To nationalize these utilities, edit the file *[sys]<sys>FontNatMsg.txt* and recreate *[sys]<sys>FontNatMsg.bin* using the Create Message File command.

Procedural Interface Changes

The procedural interface to the Font Service and the Scaling Font Service is described in the *CTOS Generic Print System Programming Guide*, which also documents the additional calls supported only by the Scaling Font Service.

Section 5
Examples of Font Tool Tasks

The examples in this section are primarily intended to show how the various elements of the Font Tool work together to complete particular tasks. If an example matches a task that you need to perform, you can use the example as step-by-step instructions. All the examples assume that you have the standard editable font database and that you are using a variant database. (For information on variant databases, see "Variant Databases," in Section 2.)

The following examples are discussed in this section:

- excluding unused font data from the runtime font database
- adding PostScript font families to the runtime font database
- modifying the HP LaserJet font keys
- adding a new HP LaserJet font cartridge
- adding HP LaserJet downloadable fonts
- adding the HP LaserJet Linedraw font
- exporting and importing font data
- nationalizing translation tables

Excluding Unused Font Data

Remember that the purpose of the Font Tool is to create a runtime font database that only includes the font data that your application programs and printers need. The best way to exclude unnecessary data from the runtime font database is to exclude unused font families and device types. That way, all the font data that pertains to those font families and device types is also excluded, for example, width tables, reduction rules, character set, and so forth.

For the sake of the following example, your only device type is an HP LaserJet. So, to exclude any extraneous font data from the runtime font database, you will

- modify the Font Families and Device Types directories
- generate a new runtime font database

To clean up the runtime font database,

1. Modify the Font Families directory, as follows:
   a. From the Font Tool main menu, open the Font Families directory.
   b. Use the arrow keys to move the highlight to a record that you want excluded from the runtime font database, and press **F5** (Modify).
   c. Use the **TAB** key to move the highlight to the *Gen?* field, and enter **No**. Press **GO**.
   d. Repeat steps **b** and **c** for every font family except Courier, Helvetica, and Times.
   e. Press **Go** to return to the main menu.
2. Modify the Device Types directory, as follows:
   a. From the Font Tool main menu, open the Device Types directory.
   b. Use the arrow keys to move the highlight to a record that you want to exclude from the runtime font database, and press **F5** (Modify).
   c. Use the **TAB** key to move the highlight to the *Gen?* field, and enter **no**. Press **GO**.
   d. Repeat steps **b** and **c** for every device type except HPLaserJet.
   e. Press **GO** to return to the main menu.

3. Generate a new runtime font database, as follows:
   a. Highlight the *Generate Runtime Database* field, and press **GO**.
   b. When the command is finished executing, exit the Font Tool and overwrite the old runtime font database with the new one. For more information on how to overwrite the runtime font database, see "Generating a New Runtime Font Database," in Section 3.

Adding PostScript Font Families

Note: *This example assumes that you have imported the additional PostScript port files into your editable font database. To do so, copy the port files from the laser printer package distribution diskettes to your hard disk, and then use the* Import *command to add them to your editable font database.*

The GPS laser printer package distribution diskettes contains port files for additional PostScript font families, including Glypha, ITC Lubalin Graph, ITC Souvenir, ITC Garamond, ITC American Typewriter, ITC Benguiat, ITC Friz Quadrata, and ITC Machine. These font families are not included in the standard runtime font database. This example shows how to add the new PostScript font families to your runtime font database.

- modify the Font Families directory
- generate a new runtime font database

To add font families to the runtime font database,

1. Modify the Font Families directory, as follows:
   a. From the Font Tool main menu, open the Font Families directory.
   b. Use the arrow keys to move the highlight to a font family that you want to add to the runtime font database, and press **F5** (Modify).
   c. Use the **TAB** key to move the highlight to the *Gen?* field, and enter **yes**.

d. Repeat steps b and c for every font family that your printer can print.

e. Press **GO** to return to the main menu.

2. Generate a new runtime font database, as follows:

a. Highlight the *Generate Runtime Database* field and press **GO**.

b. When the command is finished executing, exit the Font Tool and overwrite the old runtime font database with the new one. For more information on how to overwrite the runtime font database, see "Generating a New Runtime Font Database," in Section 3.

Modifying HP LaserJet Font Keys

Note: *This example assumes that you have imported the HP LaserJet port files into your editable font database. To do so, copy the port files from the laser printer package distribution diskettes to your hard disk, and then use the* Import *command to add them to your editable font database.*

The standard font database includes width tables for both the B and F font cartridges for the HP LaserJet. The B cartridge width tables are included in the standard runtime font database, but the ones for the F cartridge aren't. To use the F cartridge, you need to edit the font keys that use the HP LaserJet width tables. To do so, you will

- change the name of the width tables in the HP LaserJet font keys
- generate a new runtime font database

To modify the font keys,

1. Modify the font key detail forms, as follows:

a. From the Font Tool main menu, open the Font Keys directory.

b. Press **NEXT PAGE** until you find the HP LaserJet font key records.

c. Use the arrow keys to move the highlight to the record of the first proportional font, 14 point Helvetica bold.

d. Press **F5** (Modify) to display the Font Key detail form.

e. Press **TAB** to move the highlight to the *Width Table* field.

f. Press **CODE-DELETE**. Enter **HpFHelvBold14** (see Figure 5-1).

g. Press **TAB** to move the highlight to the *Alias* field. Change the font cartridge name from **92286B** to **92286F**. The escape sequence is the same for both cartridges, so leave it as is.

h. Press **GO** to return to the Font Keys directory.

i. Repeat steps d through h for the Times font keys. In each detail form, change the entries in the width table name and font cartridge name from **B** to **F**. You don't need to modify the Courier font keys because they don't use width tables.

j. Press **GO** to return to the main menu.

2. Generate a new runtime font database, as follows:

   a. Highlight the *Generate Runtime Database* field, and press **GO**.

   b. When the command is finished executing, exit the Font Tool and overwrite the old runtime font database with the new one. For more information on how to overwrite the runtime font database, see "Generating a New Runtime Font Database," in Section 3.

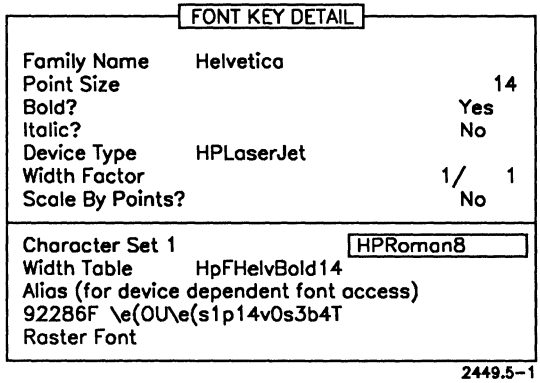


Figure 5–1. Modified HP LaserJet Font Keys

Adding an HP LaserJet Font Cartridge

The standard editable font database supports the HP LaserJet A, B, and F font cartridges. To use another cartridge, you need to add its font data to the Font Database. This example shows you how to add the T cartridge.

The HP LaserJet T cartridge can print the following fonts: 8, 10, 12, and 14 point Helvetica bold; 6 and 8 point Helvetica Roman (called *medium* in the printer documentation); and a Linedraw font. In this example, you will add the Helvetica fonts. In the next example, you will add the Linedraw font.

To add the Helvetica fonts, you will

- add width tables
- add font keys
- generate a new runtime font database

Notice that you don't need to add font families, device types, translation tables, or character sets since the Helvetica font family and the HP LaserJet printers are already in the font database. The new fonts all use the HPRoman8 character set, which is already included in the Font Database.

To add the T cartridge,

1. Create new width tables, as follows:
    a. From the Font Tool main menu, open the Width Tables directory.
    b. Press **F4** (Insert) to create a blank width table.
    c. Enter **HpTHelvBold14** in the *Name:* field.
    d. Enter **For T cartridge** in the *Comment:* field.
    e. Enter **300** and **inch** in the *Unit of Measure:* field. (The Hp LaserJet width information, also called a "spacing table," is calculated in terms of dots, which refers to the printer resolution, in this case 300 dots per inch.)

f. Enter **HPRoman8** in the *Character Set:* field. Remember that the table will now be in the order of the HP LaserJet character code set, not the CTOS character code set.

g. Using the width information in your printer documentation, enter the widths to the right of the characters (see Figure 5-2). Make sure you enter the value of the overall character cell width, also called the "cursor move" value. If a character does not exist in the HP font, leave the field blank.

CHARACTER WIDTH TABLE

Name: HpTHelvBold14 Comment: For T cartridge

Unit of Measure: 300th of a inch Character Set: HPRoman8

40	§	52	A	43	B	41	C	D	E	F	G
48	H		I		J		K	L	M	N	O
50	P		Q		R		S	T	U	V	W
58	X		Y		Z		[	\	]	^	_

2449.5-2

Figure 5–2. HpTHelvBold14 Width Table

h. Press **GO** to return to the Width Tables directory.

i. Repeat steps a through h for the rest of the Helvetica fonts. At the top of each width table, enter the same information that you did for the width table you just created, except for the name. Use names that reflect the font, for example, **HpTHelvBold8.** Make sure you include a **T** for the T cartridge.

j. Press **GO** to return to the main menu.

2. Add font keys for the new fonts, as follows:

   a. Open the Font Keys directory.

   b. Press **NEXT PAGE** until you locate the font key for HPLaserJet 14 point Helvetica bold. Highlight the record and press **F5** (Modify). The HPLaserJet 14 point Helvetica bold font key detail form is highlighted, as shown in Figure 5-3.

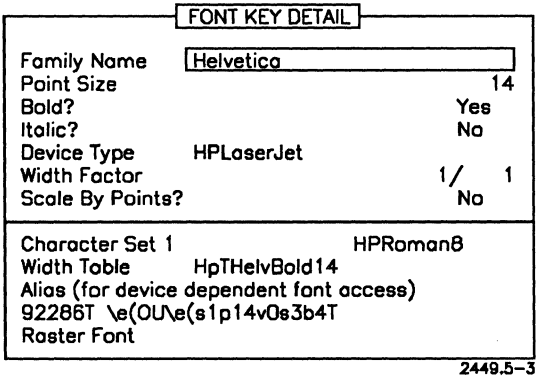


Figure 5–3. Font Key Detail Form for HP 14 Point Helvetica Bold

By modifying this font key, you include the correct width table for the T cartridge version of 14 point Helvetica bold, as opposed to the B cartridge version of 14 point Helvetica bold. If you want to use both the B and the T versions of the same font, you need to distinguish between them by creating separate font families (for example Helvetica-B and Helvetica-T), or by creating a different device type for each cartridge. For this example, you will only use one version of each font.

   c. Press **TAB** to highlight the *Width Table* field. Enter the name of the width table you created earlier, **HpTHelvBold14**. (See Figure 5-3.)

   d. Highlight the *Alias* field, and press **CODE-DELETE**. Enter the name of the cartridge, **92286T**, press **Spacebar**, and enter the following escape sequence: **\e(OU\e(s1p14v0s3b4T**

e. Press **GO** to return to the Font Keys directory.

f. Highlight the 14 point bold Helvetica key, and press **F3** (Copy). Use this detail form as a base for the rest of the font keys. Change the point size, width table name, alias, and if necessary, the entry in the *Bold?* field. (See your printer documentation for the escape sequences.)

g. Press **GO** to return to the main menu.

3. Generate a new runtime font database, as follows:

   a. Highlight the *Generate Runtime Database* field, and press **GO**.

   b. When the command is finished executing, exit the Font Tool and overwrite the old runtime font database with the new one. For more information on how to overwrite the runtime font database, see "Generating a New Runtime Font Database," in Section 3.

Adding HP LaserJet Downloadable Fonts

When you add the HP LaserJet downloadable fonts, you follow nearly the same procedure as when you add a font cartridge. For that reason, this example doesn't go into as much detail as the previous one. This example assumes that you have already copied the downloadable fonts to your hard disk.

To add HP LaserJet downloadable fonts,

1. Add font families, if necessary.
2. Create width tables for each of the new fonts. Use *HPRoman8* for the character set.
3. Create font keys for all of the new fonts, as follows:

   a. Enter **HPDownLoad** in the *Device Type* field.

   b. In the *Width Table* fields, use the names of the width tables you just created.

   c. In the *Character Set* fields, enter **HPRoman8**.

   d. Enter the font file name and the appropriate page orientation character, as explained in Appendix B.

4. Generate a new runtime font database and overwrite the old runtime font database with the new one.

5. If you are storing your downloadable fonts in a volume or directory other than *[Sys]<Gps>*, use the following format in the *Device Setup* field in the *Device* form when you use the Print Manager to install the printer:

   dir:[*VolumeName*]<*DirectoryName*>

   where *VolumeName* and *DirectoryName* are the names of the volume and directory, respectively, where you want to store the downloadable font files.

   (For for more information, see "Working With Device Types," in Section 3, and the *CTOS Generic Print System Administration Guide*.)

Remember that if your printer begins a new page whenever it prints a downloaded font, you need to return to the Device Types directory and enter **yes** the *Scan?* field.

Adding the HP LaserJet Linedraw Font

You can use the HP Linedraw font to create rules and boxes in different line weights. The Linedraw font comes with several of the HP font cartridges, but for this example you'll use the T cartridge.

When you add the Linedraw font, you won't add a Linedraw font family. Instead, you'll add the font as an alternate character set for font keys based on other font families. This way, you can use the Linedraw characters without having to change fonts from the Document Designer *Fonts* menu. For more information on alternate character sets, see "Alternate Character Sets," in Section 2.

To add the Linedraw font, you will follow these basic steps:

- create a new translation table
- create a new character set
- add the new character set to the 12 point Courier font key as an alternate character set

To add the Linedraw font,

1. Create a new translation table, as follows:
   a. From the Font Tool main menu, open the Translation Tables directory.
   b. Press **F4** (Insert) to create a new translation table.
   c. Enter **CtToHpSymbols** in the *Name:* field (see Figure 5-4).
   d. Enter **For Linedraw font** in the *Comment:* field.
   e. Enter **CT** in the *Character Set:* field.
   f. Highlight the *Translates to Character Set:* field, and press **HELP** to display the Character Sets directory.
   g. Press **F4** (Insert) to create a New Character Set detail form.
   h. Enter **HPSymbols** in the *Name:* field.
   i. Press **GO** to return to the Character Sets directory. Press **GO** again to return to the Translation Table. The name *HpSymbols* is entered in the *Translates to Character Set:* field.
   j. Enter the hexadecimal codes in the translation table in the range C0h to FFh. You'll be able to use most of the symbols in the Linedraw font, but not all of them. It's easiest to enter the codes that match the symbols in your printer documentation with the symbols that are displayed on the monitor. For example, the symbol at the C0h position in the translation table in Figure 5-4 looks identical to the symbol that maps to the decimal code 33 in the HP character code set. Since 33 decimal converts to 21 hexadecimal, enter 21 next to the character at the C0h position (see Figure 5-4).
   k. Press **GO** twice to return to the main menu.

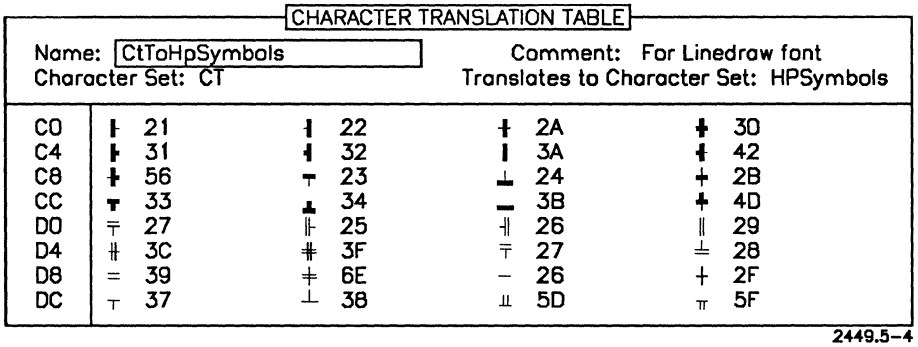

CHARACTER TRANSLATION TABLE
Name: CtToHpSymbols
Character Set: CT
Comment: For Linedraw font
Translates to Character Set: HPSymbols
C0 21 22 2A 30
C4 31 32 3A 42
C8 56 23 24 2B
CC 33 34 3B 4D
D0 27 25 26 29
D4 3C 3F 27 28
D8 39 6E 26 2F
DC 37 38 5D 5F
2449.5-4

Figure 5–4. Linedraw Translation Table

2. Finish the new character set, as follows:

   a. Open the Character Sets directory.

   b. Highlight the HPSymbols character set, and press **F5** (Modify). The HPSymbols character set detail form is displayed.

   c. Enter **For Linedraw font** in the *Comment:* field.

   d. Enter **CtToHpSymbols** in the *From Ct:* field (see Figure 5-5).

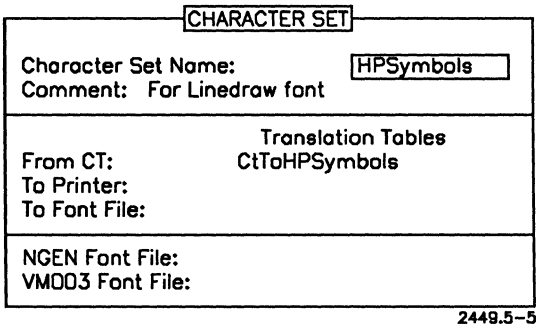


Figure 5–5. HPSymbols Character Set

   e. Press **GO** twice to return to the main menu.

3. Modify font keys, as follows:

   a. Open the Font Keys directory.

   b. Highlight the HP LaserJet 12 point Courier roman font key record, and press **F5** (Modify).

   c. Press **NEXT PAGE**. Enter **HPSymbols** in the *Character Set 2* field (see Figure 5-6).

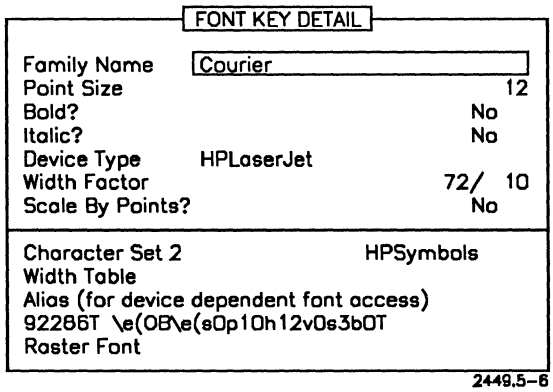


Figure 5–6. Linedraw Character Set in Courier Font Keys

   d. Enter the name of the cartridge and the Linedraw escape sequence in the *Alias* field, as shown below:

      92286T \e(OB\e(s0p10h12v0s3b0T

      Make sure you insert a space between the cartridge name and the escape sequence.

   e. Press **GO** to return to the Font Keys directory. Repeat steps b through d for the 12 point Courier italic and 12 point Courier bold HP LaserJet font keys.

   f. Press **GO** twice to return to the main menu.

4. Generate a new runtime font database, as follows:

   a. Highlight the *Generate Runtime Database* field, and press **GO**.

b. When the command is finished executing, exit the Font Tool and overwrite the old runtime font database with the new one. For more information on how to overwrite the runtime font database, see "Generating a New Runtime Font Database," in Section 3.

Exporting and Importing Data

This example shows you how to export and import all the data for a single device type. Let's say you've added the device type SuperPrinter to your editable font database, which means you've added all the SuperPrinter font families, character sets, width tables, font keys, reduction rules, and so forth. You now want to distribute the SuperPrinter font data to all the other editable font databases in your company. To do so, you will

- export all SuperPrinter records to port files
- copy the port files to the importing FontDb directory
- import the data from the port files

To export and import the data for one device type,

1. In the Executive command line, type **Font Tool**, and press **RETURN**. To enter the variant database SuperPrinter, fill in the command form, as shown below.

   Font Tool
   [Batch mode commands] ExportAllVariant
   [Variant database] SuperPrinter

2. Press **GO**. All the font data pertaining to the SuperPrinter device type is copied into port files in the *<FontDb>* directory.
3. Copy the port files into the importing *<FontDb>* directory. (You can do this by using a floppy disk or by using the master workstation in a cluster.)

4. In the Executive, enter **Font Tool** in the command line, and press **RETURN**. Fill in the command form, as shown below. In this example, the port files for SuperPrinter are being imported into a previously created variant database called MaxFonts.

```
Font Tool
   [Batch mode commands]     Import:SuperPrinter
   [Variant database]        MaxFonts
```

   To be able to use the SuperPrinter font data once it has been imported into the MaxFonts variant database, you need to generate a runtime font database from the MaxFonts variant database.

Nationalizing Translation Tables

When you nationalize a character set, your work with the Font Tool is only part of the overall procedure. This example assumes that you have already added the new character code set to your system. That is to say, your system font should be nationalized already. (For more information, see your Font Designer manual and the discussions of keyboard assignments and native language support in the *CTOS Operating System Concepts Manual.*)

In the nationalization process, you use the Font Tool to create new translation tables that are based on the nationalized character code set. The character codes A0h-B5h, 83h-87h, B7h-BCh, BEh, and BFh have been reserved for nationalization. Since this range is somewhat limited, it is common to create character code sets for specific languages or groups of languages, for example, for Yugoslavian or "Scandinavian."

In this example, you will nationalize a character set for Andorra by creating a new character code set for Andorra, called "Andorra." To do so, you will change all the first level printer translation tables to translate from the Andorra character code set instead of the CTOS character code set. For example, instead of *CtToHPRoman8*, you will have *AndorraToHPRoman8*.

To make it easier to understand, this example is unrealistically simple. Usually you would be adding or changing the character codes for several characters.

To nationalize a character code set for Andorra, you will

- Create a new variant database called Andorra.
- Create a new character set called Andorra.
- Create a translation table that translates from the standard CTOS character code set to the Andorra character code set. This translation table will be used to shuffle the first level printer translation tables into the order of the Andorra character code set.
- Shuffle the first level translation tables and add any new characters that are not in the CTOS character code set.
- Modify the character sets that include the modified translation tables.
- Generate a new runtime font database.

When you nationalize a character code set, you may use some of the characters in the CTOS character code set and you may add new ones. You may also change the character codes on characters that are already in the CTOS character code set.

For the sake of this example, the character code set for Andorra has two characters that are different from the CTOS character code set: an *e* with an acute accent (*é*), and an *e* ring (*e̊*). Each character presents a different problem.

- The *é* is in the CTOS character set, but is represented by a different character code. The CTOS character code for *é* is AC, but the Andorra character code for *é* is A8h. Therefore, you need to change all first level translation tables so that the *é* is in the A8 location.
- The *e̊* is not in the CTOS character code set, so you need to add it to all the first level translation tables.

For this example, it is assumed that you have already added the Andorra character code set to your system and that the Andorra character code set is now the system font. It is also assumed that you have saved a copy of the standard system font for high-resolution monitors, *1024x768_80Sys.font*, or the standard system font for character-mapped monitors, *T1Sys.font*. If you don't have a copy, you can get one from the Standard Software distribution diskettes, or you can create new files using the font files *Save>1024x768_80Sys.font* and *Save>T1Sys.font*, which are located in *[Sys]<Sys>*.

To nationalize the translation tables,

1. Create the variant database Andorra, as follows:
   a. In the Executive command line, type **Font Tool**, and press **RETURN**.
   b. Fill in the command form, as shown below.

      Font Tool
      [Batch mode commands] Create
      [Variant database] Andorra

   c. Press **GO**.
2. Enter the Font Tool, as follows:
   a. In the Executive command line, type **Font Tool**, and press **RETURN**.
   b. Fill in the command form as shown below.

      Font Tool
      [Batch mode commands]
      [Variant database] Andorra

   c. Press **GO**.
3. Create a new character set, as follows:
   a. From the Font Tool main menu, open the Character Sets directory.
   b. Press **F4** (Insert) to create a new character set. Enter **Andorra** in the *Name:* field, and press **GO**.
   c. Press **GO** to return to the Character Sets directory.
4. Modify the CTOS character set, as follows:
   a. Highlight the CTOS character set, and press **F5** (Modify).
   b. In the *Ngen Font File:* field, enter **T1Sys.font**. If you are using a high resolution monitor, enter **1024x768_80Sys.font** in the *VM-003 Font File:* field. If you leave the field blank, the translation tables will be shown in the order of the system font, in this case the Andorra font, which may get confusing. Press **GO** twice to return to the main menu.

5. Create the *CtToAndorra* translation table, as shown below. This maps the CTOS character code set to the Andorra character code set.

   a. Open the Translation Tables directory.

   b. Highlight the *CTToAscii* record, and press **F3** (Copy). A copy of *CTToAscii* is displayed.

   c. Change the name from **CTToAscii** to **CtToAndorra** (see Figure 5-7).

CHARACTER TRANSLATION TABLE

Name: CtToAndorra Comment:
Character Set: CT Translates to Character Set: Andorra

A0	Å A0	ð A1	Ä A2	ö A3
A4	Ö A4	ö A5	Ø A6	ø A7
A8	Ü	ü A9	ç AA	ê AB
AC	é A8	è AD	ff AE	æ AF
B0	ß B0	£	° B2	© B3
B4	® B4	™ B5	¹	¹₁
B8	¹₂	¹₃	¹₄	¹₅
BC	¹₆	¹₇	¹₈	¹₉

2449.5-7

Figure 5–7. CtToAndorra Translation Table

   d. Enter **Andorra** in the *Translates to Character Set:* field.

   e. Press **NEXT PAGE** three times to display the range from A0h to BFh.

   f. Enter the Andorra character codes for all the characters in the range A0h through B5h. In this case, all the Andorra character codes are the same as the CTOS character codes except the following two:

      - The *é* must be mapped from ACh to A8h, so enter **A8** at the ACh location in the table (see Figure 5-7).
      - The ℓ is not included in the CTOS character code set, so it can't be mapped with this translation table. The Andorra character code for the ℓ is B1h, which is the CTOS character code for the pound symbol (£). The pound symbol is not part of the Andorra character code set, so leave the B1h location blank.

If you don't modify the CTOS character set as explained in step 4, the characters might look like they are in the wrong places. If this is the case, use Appendix A to find the character codes for the CTOS character code set.

g. Press **GO** to return to the Translation Tables directory.

6. Shuffle the first level translation tables. For this example, you will only shuffle one, but the procedure is the same for all the tables. For the sake of this example, a fictional printer, SuperPrinter, will be used. The SuperPrinter character code for the *é* is C5h. The SuperPrinter character code for the *ê* is 83h.

   a. Highlight the record for the *CtToSuperPrinter* translation table, and press **F3** (Copy).

   b. Enter **AndorraToSuperPrinter** in the *Name:* field.

   c. Enter **Andorra** in the *Character Set:* field.

   d. Press **NEXT PAGE** three times until you locate the A0h-BFh range. Notice that the translation table character ordering is now based on the order of the Andorra character code set; *é* is now at the A8h location. However, the SuperPrinter character code for *é*, C5h, is in the wrong location. It's still at the location it was mapped to in the CTOS character code set. However, when you shuffle the table, the C5h code will move to the A8h position.

   e. Press **F10** (Shuffle) to display the Shuffle Table, as shown in Figure 5-8.

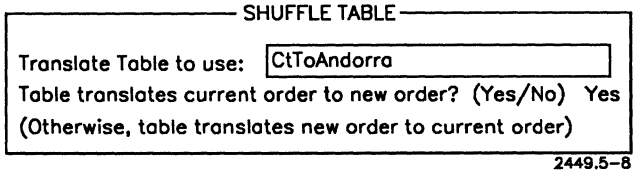

SHUFFLE TABLE
Translate Table to use: CtToAndorra
Table translates current order to new order? (Yes/No) Yes
(Otherwise, table translates new order to current order)
2449.5–8

Figure 5–8. Shuffle Table

f. Enter **CtToAndorra** in the *Translate Table to use:* field of the Shuffle Table, and press **GO**.

The translation table is shuffled into the order of the Andorra character code set, which changes the mapping for the SuperPrinter é. In the CTOS character code set, it was ACh to C5h. Now, in the Andorra character code set, it is A8h to C5h. Figure 5-9 shows the translation table before and after it is shuffled. Notice that the SuperPrinter code doesn't change; it just moves to a new location in the table.

BEFORE SHUFFLE:

CHARACTER TRANSLATION TABLE

Name: AndorraToSuperPrinter Comment:
Character Set: Andorra Translates to Character Set: SuperPrinter

A0	Å D0	ð D4	Ä D8	ď CC
A4	Ö DA	ŏ CE	Ø D2	ø D6
A8	é	ü CF	ç B5	ê C1
AC	C5	è C9	Æ D3	æ D7
B0	ß DE	ê	° B3	© A1
B4	® A2	™ A3	'	'
B8	'	'	'	'
BC	'	'	'	'

AFTER SHUFFLE:

CHARACTER TRANSLATION TABLE

Name: AndorraToSuperPrinter Comment:
Character Set: Andorra Translates to Character Set: SuperPrinter

A0	Å D0	ð D4	Ä D8	ď CC
A4	Ö DA	ŏ CE	Ø D2	ø D6
A8	é C5	ü CF	ç B5	ê C1
AC		è C9	Æ D3	æ D7
B0	ß DE	ê	° B3	© A1
B4	® A2	™ A3	'	'
B8	'	'	'	'
BC	'	'	'	'

2449.5-9

Figure 5–9. Shuffling the Translation Tables

7. Add the character code for ℓ, as follows:

   a. Since ℓ is not included in the CTOS character code set, you need to add it to the AndorraToSuperPrinter translation table. The Andorra character code for ℓ is B1h, so there is a blank place in the table. Enter the printer character code for ℓ in the B1h location, in this case **83**. Figure 5-10 shows the completed translation table.

   b. Press **GO** twice to return to the main menu.

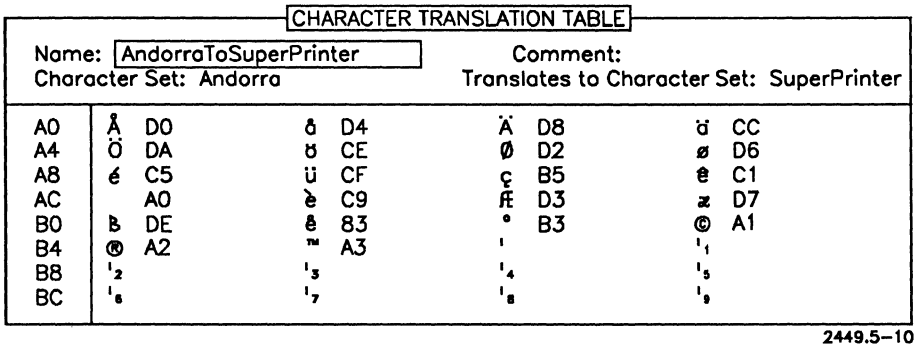

CHARACTER TRANSLATION TABLE
Name: AndorraToSuperPrinter
Comment:
Character Set: Andorra
Translates to Character Set: SuperPrinter
A0 Å D0 ð D4 Ä D8 ď CC
A4 Ö DA ŏ CE Ø D2 ø D6
A8 é C5 ü CF ç B5 ê C1
AC A0 è C9 Æ D3 æ D7
B0 ß DE ê 83 ° B3 © A1
B4 ® A2 ™ A3 ' '1
B8 '2 '3 '4 '5
BC '6 '7 '8 '9
2449.5–10

Figure 5–10. Completed Translation Table

8. Modify the SuperPrinter character set, as follows:

   a. Open the Character Sets directory.

   b. Highlight the SuperPrinter character set record, and press **F5** (Modify).

   c. Enter **AndorraToSuperPrinter** in the *From Ct:* field, and press **GO**.

   d. Press **GO** again to return to the main menu.

9. If you were adding fonts that used the new character code set, you would need to add the character widths for any new characters. In this example, the new character is an accented *e*, so the width will be the same as all the other *e* characters.

10. Generate a new runtime font database, as follows:

    a. Highlight the *Generate Runtime Database* field, and press **GO**.

    b. When the command is finished executing, exit the Font Tool and overwrite the old runtime font database with the new one. For more information on how to overwrite the runtime font database, see "Generating a New Runtime Font Database," in Section 3.

Note: *In this last example, if you had been adding all new characters, you would not have used the* Shuffle *command, since there would not have been anything to shuffle.*

Appendix A
CTOS and Screenfont Character Code Sets

This appendix shows the standard CTOS character code set and the Screenfont Database character code set. Use the Screenfont character code set when you modify raster font translation tables. Use the CTOS character code set when you create or modify printer font translation tables.

Table A-1. Graphic Representation of the Standard Character Set

Character Code	Video Display Character	Character Code	Video Display Character	Character Code	Video Display Character	Character Code	Video Display Character
00h		20h		40h	@	60h	`
01h	↑	21h	!	41h	A	61h	a
02h	◣	22h	"	42h	B	62h	b
03h	¢	23h	#	43h	C	63h	c
04h	■	24h	$	44h	D	64h	d
05h	□	25h	%	45h	E	65h	e
06h	½	26h	&	46h	F	66h	f
07h	⚠	27h	'	47h	G	67h	g
08h	⇤	28h	(	48h	H	68h	h
09h	⇥	29h	)	49h	I	69h	i
0Ah	♪	2Ah	*	4Ah	J	6Ah	j
0Bh	↓	2Bh	+	4Bh	K	6Bh	k
0Ch	\	2Ch	,	4Ch	L	6Ch	l
0Dh	◢	2Dh	–	4Dh	M	6Dh	m
0Eh	←	2Eh	.	4Eh	N	6Eh	n
0Fh	‡	2Fh	/	4Fh	O	6Fh	o
10h	¼	30h	0	50h	P	70h	p
11h	↕	31h	1	51h	Q	71h	q
12h	→	32h	2	52h	R	72h	r
13h	⌙	33h	3	53h	S	73h	s
14h		34h	4	54h	T	74h	t
15h	÷	35h	5	55h	U	75h	u
16h	ı	36h	6	56h	V	76h	v
17h	$	37h	7	57h	W	77h	w
18h	≠	38h	8	58h	X	78h	x
19h	≃	39h	9	59h	Y	79h	y
1Ah	¶	3Ah	:	5Ah	Z	7Ah	z
1Bh	•	3Bh	;	5Bh	[	7Bh	{
1Ch	–	3Ch	<	5Ch	\	7Ch	|
1Dh	≤	3Dh	=	5Dh	]	7Dh	}
1Eh	±	3Eh	>	5Eh	^	7Eh	~
1Fh	≥	3Fh	?	5Fh	_	7Fh	■

501AA-1.1

continued

Table A-1. Graphic Representation of the Standard Character Set (cont.)

Character Code	Video Display Character	Character Code	Video Display Character	Character Code	Video Display Character	Character Code	Video Display Character
80h		A0h	Å	C0h	┠	E0h	┖
81h	|	A1h	å	C1h	┨	E1h	┃
82h	|0	A2h	Ä	C2h	╂	E2h	┠
83h	|1	A3h	ä	C3h	╋	E3h	┨
84h	|2	A4h	Ö	C4h	┣	E4h	┒
85h	|3	A5h	ö	C5h	┫	E5h	┣
86h	|4	A6h	Ø	C6h	┃	E6h	┫
87h	|5	A7h	ø	C7h	╋	E7h	┛
88h	|6	A8h	Ü	C8h	╋	E8h	┓
89h	|7	A9h	ü	C9h	┳	E9h	┗
8Ah	|8	AAh	ç	CAh	┻	EAh	┛
8Bh	|9	ABh	ê	CBh	╋	EBh	┏
8Ch	0	ACh	é	CCh	┳	ECh	┓
8Dh	1	ADh	è	CDh	┻	EDh	└
8Eh	2	AEh	Æ	CEh	━	EEh	┘
8Fh	3	AFh	æ	CFh	╋	EFh	┌
90h	4	B0h	ß	D0h	╋	F0h	┐
91h	5	B1h	£	D1h	╟	F1h	╟
92h	6	B2h	°	D2h	╢	F2h	╥
93h	7	B3h	©	D3h	╢	F3h	╢
94h	8	B4h	®	D4h	╫	F4h	╨
95h	9	B5h	™	D5h	╫	F5h	╞
96h	0	B6h	'	D6h	╥	F6h	╥
97h	1	B7h	'1	D7h	╨	F7h	╡
98h	2	B8h	'2	D8h	═	F8h	╨
99h	3	B9h	'3	D9h	╪	F9h	╞
9Ah	4	BAh	'4	DAh	─	FAh	╥
9Bh	5	BBh	'5	DBh	┼	FBh	╡
9Ch	6	BCh	'6	DCh	┬	FCh	╨
9Dh	7	BDh	'7	DDh	┴	FDh	▨
9Eh	8	BEh	'8	DEh	┴	FEh	▨
9Fh	9	BFh	'9	DFh	┬	FFh	▩

501A.A-1.2

Table A–2. Standard Character Set

Character Code	Video Display Character	Literal Keystroke(s)
00h	null	Help
01h	↑ (up arrow)	Up arrow
02h	(triangle in non-WYSIWYG mode)	Mark
03h	¢	Shift-6
04h	filled square	Finish
05h	empty square	Prev Page
06h	1/2	1/2
07h	bell	Cancel
08h	backspace	Backspace
09h	tab	Tab
0Ah	new line	Return Next
0Bh	↓ (down arrow)	Down arrow
0Ch	formfeed	Next Page
0Dh	triangle	Bound
0Eh	← (left arrow)	Left arrow
0Fh	‡ (double dagger)	Move
10h	1/4	Shift-1/2
11h	†	Scroll Up
12h	→ (right arrow)	Right arrow
13h	trough	Scroll Down
14h	raised dot	Copy
15h	÷	F1
16h	| (vertical bar)	F2
17h	§	F3

continued

Table A–2. Standard Character Set (cont.)

Character Code	Video Display Character	Literal Keystroke(s)
18h	≠	F4
19h	similarity	F5
1Ah	¶	F6
1Bh	• (bullet)	Go
1Ch	not	F7
1Dh	≤	F8
1Eh	±	F9
1Fh	≥	F10
20h	space	Spacebar
21h	!	Shift-!
22h	"	Shift-'
23h	#	Shift-3
24h	$	Shift-4
25h	%	Shift-5
26h	&	Shift-7
27h	'	'
28h	(	Shift-9
29h	)	Shift-0
2Ah	*	Shift-8
2Bh	+	Shift-=
2Ch	, (comma)	,
2Dh	- (hyphen)	-
2Eh	. (period)	.
2Fh	/	/
30h	0	0

Table A–2. Standard Character Set (cont.)

Character Code	Video Display Character	Literal Keystroke(s)
31h	1	1
32h	2	2
33h	3	3
34h	4	4
35h	5	5
36h	6	6
37h	7	7
37h	8	8
39h	9	9
3Ah	:	Shift-;
3Bh	;	;
3Ch	<	Shift-[
3Dh	=	=
3Eh	>	Shift-]
3Fh	?	Shift-/
40h	@	Shift-2
41h	A	Shift-a
42h	B	Shift-b
43h	C	Shift-c
44h	D	Shift-d
45h	E	Shift-e
46h	F	Shift-f
47h	G	Shift-g
48h	H	Shift-h
49h	I	Shift-i

Table A–2. Standard Character Set (cont.)

Character Code	Video Display Character	Literal Keystroke(s)
4Ah	J	Shift-j
4Bh	K	Shift-k
4Ch	L	Shift-l
4Dh	M	Shift-m
4Eh	N	Shift-n
4Fh	O	Shift-o
50h	P	Shift-p
51h	Q	Shift-q
52h	R	Shift-r
53h	S	Shift-s
54h	T	Shift-t
55h	U	Shift-u
56h	V	Shift-v
57h	W	Shift-w
58h	X	Shift-x
59h	Y	Shift-y
5Ah	Z	Shift-z
5Bh	[	[
5Ch	\ (backslash)	Shift-num 8
5Dh	]	]
5Eh	^ (caret)	^
5Fh	_ (underline)	Shift-- (hyphen)
60h	' (reverse accent)	Shift-num 1
61h	a	a
62h	b	b

Table A–2. Standard Character Set (cont.)

Character Code	Video Display Character	Literal Keystroke(s)
63h	c	c
64h	d	d
65h	e	e
66h	f	f
67h	g	g
68h	h	h
69h	i	i
6Ah	j	j
6Bh	k	k
6Ch	l	l
6Dh	m	m
6Eh	n	n
6Fh	o	o
70h	p	p
71h	q	q
72h	r	r
73h	s	s
74h	t	t
75h	u	u
76h	v	v
77h	w	w
78h	x	x
79h	y	y
7Ah	z	z
7Bh	{	Shift-num 4

Table A–2. Standard Character Set (cont.)

Character Code	Video Display Character	Literal Keystroke(s)
7Ch	|	Shift-num 7
7Dh	}	Shift-num 5
7Eh	~ (tilde)	Shift-^
7Fh	filled rectangle	Delete
A0h	A circle	Code-Spacebar
A1h	a circle	Code-Shift-1
A2h	A umlaut	Code-Shift-'
A3h	a umlaut	Code-Shift-3
A4h	O umlaut	Code-Shift-4
A5h	o umlaut	Code-Shift-5
A6h	O slashed	Code-Shift-7
A7h	o slashed	Code-'
A8h	U umlaut	Code-Shift-9
A9h	u umlaut	Code-Shift-0
AAh	c cedilla	Code-Shift-8
ABh	e circumflex	Code-Shift-=
ACh	e grave	Code-,
ADh	e acute	Code-- (hyphen)
AEh	AE dipthong	Code-.
AFh	ae dipthong	Code-/
B0h	ß	Code-0
B1h	£	Code-1
B2h	° (degree)	Code-2
B3h	©	Code-3

Table A–2. Standard Character Set (cont.)

Character Code	Video Display Character	Literal Keystroke(s)
B5h	™	Code-5
B4h	®	Code-4
B6h	¡	Code-6
B7h	¿	Code-7
B8h	"	Code-8
B9h	"	Code-9
BAh	'	Code-Shift-;
BBh	„	Code-;
BCh	‹	Code-Shift-[
BDh	›	Code-=
BEh	–	Code-Shift-]
BFh	—	Code-Shift-/
COh	see Table A-1	Code-Shift-2 Shift-Help
C1h	see Table A-1	Code-Shift-a Shift-Up arrow
C2h	see Table A-1	Code-Shift-b Shift-Mark
C3h	see Table A-1	Code-Shift-c Shift-Bound
C4h	see Table A-1	Code-Shift-d Shift-Finish
C5h	see Table A-1	Code-Shift-e Shift-Prev Page
C6h	see Table A-1	Code-Shift-f
C7h	see Table A-1	Code-Shift-g Shift-Cancel

Table A–2. Standard Character Set (cont.)

Character Code	Video Display Character	Literal Keystroke(s)
C8h	see Table A-1	Code-Shift-h Shift-Delete
C9h	see Table A-1	Code-Shift-i Shift-Go
CAh	see Table A-1	Code-Shift-j Shift-F9
CBh	see Table A-1	Code-Shift-k Shift-Down arrow
CCh	see Table A-1	Code-Shift-l Shift-Next Page
CDh	see Table A-1	Code-Shift-m Shift-F8
CEh	see Table A-1	Code-Shift-n Shift-Left arrow
CFh	see Table A-1	Code-Shift-o Shift-Move
D0h	see Table A-1	Code-Shift-p OVERTYPE
D1h	see Table A-1	Code-Shift-q Shift-Scroll Up
D2h	see Table A-1	Code-Shift-r Shift-Right arrow
D3h	see Table A-1	Code-Shift-s Shift-Scroll Down
D4h	see Table A-1	Code-Shift-t Shift-Copy
D5h	see Table A-1	Code-Shift-u Shift-F1
D6h	see Table A-1	Code-Shift-v Shift-F2

Table A–2. Standard Character Set (cont.)

Character Code	Video Display Character	Literal Keystroke(s)
D7h	see Table A-1	Code-Shift-w Shift-F3
D8h	see Table A-1	Code-Shift-x Shift-F4
D9h	see Table A-1	Code-Shift-y Shift-F5
DAh	see Table A-1	Code-Shift-z Shift-F6
DBh	see Table A-1	Code-[
DCh	see Table A-1	Code-Shift-num 8 Shift-F8
DDh	see Table A-1	Code-Shift-]
DEh	see Table A-1	Code-^
DFh	see Table A-1	Code-Shift-- (hyphen) Shift-F10
E0h	see Table A-1	Code-Shift-num 1
E1h	see Table A-1	Code-a
E2h	see Table A-1	Code-b
E3h	see Table A-1	Code-c
E4h	see Table A-1	Code-d
E5h	see Table A-1	Code-e
E6h	see Table A-1	Code-f
E7h	see Table A-1	Code-g
E8h	see Table A-1	Code-h
E9h	see Table A-1	Code-i
EAh	see Table A-1	Code-j
EBh	see Table A-1	Code-k

Table A–2. Standard Character Set (cont.)

Character Code	Video Display Character	Literal Keystroke(s)
ECh	see Table A-1	Code-l
F2h	see Table A-1	Code-r
EDh	see Table A-1	Code-m
EEh	see Table A-1	Code-n
EFh	see Table A-1	Code-o
F0h	see Table A-1	Code-p
F1h	see Table A-1	Code-q
F3h	see Table A-1	Code-s
F4h	see Table A-1	Code-t
F5h	see Table A-1	Code-u
F6h	see Table A-1	Code-v
F7h	see Table A-1	Code-w
F8h	see Table A-1	Code-y
F9h	see Table A-1	Code-x
FAh	see Table A-1	Code-z
FBh	see Table A-1	Code-Shift-num 4
FCh	see Table A-1	Code-Shift-num 7
FDh	bar chart	Code-Shift-num 5
FEh	bar chart	Code-Shift-^
FFh	bar chart	Code-Delete

Table A-3 shows the Screenfont Database character ordering used by most of the raster font files distributed with the standard editable font database. The equivalent CTOS character code, if any, is also shown. Note that many of the characters in the Screenfont Database are accented characters required for nationalization. In many cases, these characters have no equivalent characters in the CTOS character code set.

Table A-3. Screenfont Database Character Code Set

Screenfont Code	Character	CTOS Code
01	Baseline double quote („)	BB
02	Left double quote (")	B8
03	Left angle quote	
04	Florin; function	
5	Per mil	
06	Open box	05
0E	Leader dot	
0F	Double dagger (‡)	0F
11	Single dagger (†)	11
12	Right double quote (")	B9
13	Right angle quote	
15	Division (÷)	15
16	Multiplication	
17	Vertical quote (')	BA
19	Tilde (~)	BE
1B	Bullet (•)	1B
1C	En dash	
1D	Em dash (—)	BF
1E	Trademark (™)	B5

continued

Table A-3. Screenfont Database Character Code Set (cont.)

Screenfont Code	Character	CTOS Code
1F	Fraction bar	
20	Space	20
21	!	21
22	"	22
23	#	23
24	$	24
25	%	25
26	&	26
27	Right single quote (')	27
28	(	28
29	)	29
2A	*	2A
2B	+	2B
2C	Comma (,)	2C
2D	Minus sign (–)	2D
2E	.	2E
2F	/	2F
30 through 39	0 through 9	0 through 39
3A	:	3A
3B	;	3B
3C	<	3C
3D	=	3D
3E	>	3E
3F	?	3F
40	@	40

Table A-3. Screenfont Database Character Code Set (cont.)

Screenfont Code	Character	CTOS Code
41 through 5A	A through Z	41 through 5A
5B	[	5B
5C	\	5C
5D	]	5D
5E	Circumflex accent (^)	5E
5F	Underscore (_)	5F
60	Left single quote (')	60
61 through 7A	a through z	61 through 7A
7B	{	7B
7C	|	7C
7D	}	7D
7E	Tilde (accent)	7E
80	ff ligature	
81	ffi ligature	
82	ffl ligature	
83	fi ligature	
84	fl ligature	
85	Dotless i	
88	Lower grave accent	
89	Lower circumflex	
8A	Lower tilde accent	
8B	Lower ring accent	
8C	Lower breve accent	
8D	Lower caron accent	

Table A-3. Screenfont Database Character Code Set (cont.)

Screenfont Code	Character	CTOS Code
8E	Upper grave accent	
8F	Horizontal bar	
A1	Inverted exclamation	
A2	Cent (¢)	03
A5	Yen	
A3	Pound sterling (£)	B1
A7	Section mark (§)	17
A8	Diaeresis	
A9	Copyright (©)	B3
AB	Left double angle quote	
AD	Hyphen	
AE	Registered trademark (®)	B4
AF	Macron accent	
B1	Ring, degree (°)	B2
B4	Lower acute accent	
B6	Pilcrow, paragraph (¶)	1A
B7	Dot accent	
B8	Cedilla	
BB	Right double angle quote	
BC		10
BD		06
BE		02
C0	A with grave accent	
C1	A with acute accent	
C2	A with circumflex	

Table A-3. Screenfont Database Character Code Set (cont.)

Screenfont Code	Character	CTOS Code
C3	A with tilde	
C4	A with diaeresis (Ä)	A2
C5	A with ring accent (Å)	A0
C6	AE dipthong (Æ)	AE
C7	C with cedilla	
C8	E with grave accent	
C9	E with acute accent	
CA	E with circumflex	
CB	E with diaeresis	
CC	I with grave accent	
CD	I with acute accent	
CE	I with circumflex	
CF	I with diaeresis	
D0	Reserved	
D1	N with tilde	
D2	O with grave accent	
D3	O with acute accent	
D4	O with circumflex	
D5	O with tilde	
D6	O with diaeresis (Ö)	A4
D7	OE dipthong	
D8	O with slash (Ø)	A6
D9	U with grave accent	
DA	U with acute accent	

Table A-3. Screenfont Database Character Code Set (cont.)

Screenfont Code	Character	CTOS Code
DB	U with circumflex	
DC	U with diaeresis (Ü)	A8
DD	Y with acute accent	
DE	Reserved	
DF	German sharp s (ß)	B0
E0	a with grave accent	
E1	a with acute accent	
E2	a with circumflex	
E3	a with tilde	
E4	a with diaeresis (ä)	A3
E5	a with ring accent (å)	A1
E6	ae dipthong (æ)	AF
E7	c with cedilla (ç)	AA
E8	e with grave accent (è)	AD
E9	e with acute accent (é)	AC
EA	e with circumflex (ê)	AB
EB	e with diaeresis	
EC	i with grave accent	
ED	i with acute accent	
EE	i with circumflex	
EF	i with diaeresis	
F0	Reserved	
F1	n with tilde	
F2	o with grave accent	

Table A-3. Screenfont Database Character Code Set (cont.)

Screenfont Code	Character	CTOS Code
F3	o with acute accent	
F4	o with circumflex	
F5	o with tilde	
F6	o with diaeresis (ö)	A5
F7	oe dipthong	
F8	o with slash (ø)	A7
F9	u with grave accent	
FA	u with acute accent	
FB	u with circumflex	
FC	u with diaeresis (ü)	A9
FD	y with acute accent	
FE	Reserved	
FF	y with diaeresis	

Appendix B
Aliases

This appendix describes how each GPS supported device driver or printer uses the *Alias* field in the font key. For a description of the *Alias* field, see "Font Keys," in Section 2. For more information about your printer, see the GPS 2.3 Release Notice and your printer documentation.

Binary Device Driver

The binary mode driver does not use the Font Database.

CT-2400 Printers

The CT-2400 driver reads the *Alias* field to find the keywords that indicate the necessary escape sequences to send to the printer in order to print the appropriate font. These keywords are:

Bold. The font is printed bold.

Italic. The font is printed italic.

Compressed. The font is compressed horizontally.

Double. The font is printed double height and width.

Pica. The font is printed in 10 pitch, unless it is doubled or compressed.

Elite. The font is printed in 12 pitch, unless it is doubled or compressed.

PS. Selects proportional spacing.

Font1, Font2, etc. Selects the choice of resident fonts on the printer. (There is no *Font0.*)

Card. Selects a font on a font cartridge.

You can enter more than one keyword, but certain combinations may result in one or more of the keywords being ignored. For example, a font can't be both doubled and compressed at the same time.

The *Alias* field contains the following entry for the CT-2400 24 point Courier bold italic font key:

```
Double, Bold, Italic
```

(Notice that keywords are separated by a comma and a space.)

Daisy Printers

The daisy driver reads the *Alias* field to determine which print wheel or thimble the font uses and to find and transmit any escape sequences needed by the printer. The name of the print wheel is displayed in the *Alias* field, as follows:

```
Courier 72
```

You can use the daisy device driver to control an HP LaserJet printer in order to emulate a daisy device. In that case, the daisy driver reads the *Alias* field to determine which font cartridge to use and to transmit any escape sequences needed by the printer. The *Alias* field contains the following entry: the name of the cartridge, a space, and the escape sequences, as follows:

```
CartridgeA \eABC\eDEF
```

(Notice that there are no spaces between the escape sequences.)

Epson Printers

The Epson Fx-286 driver reads the *Alias* field for keywords, which are used to apply attributes to the fonts. The keywords include *Bold, Italic, Pica, Elite, Draft, Nlq* (near letter quality), *Wide*, and *Cond* (condensed). You can enter more than one keyword, but certain combinations may result in one or more of the keywords being ignored. For example, a font can't be both wide and condensed at the same time.

The *Alias* field contains the following keywords for the Epson 20 point Courier italic font key:

```
Elite, Draft, Wide, Italic
```

(Notice that keywords are separated by a comma and a space.)

HP LaserJet Printers

Since the HPLaserJet, LZ-05, LZ-12, and LZ-18 device types all use the HP LaserJet driver, they all read the Alias field the same way, that is, the driver reads the *Alias* field for font cartridge names and escape sequences.

Enter the name of the font cartridge, enter a space, and then enter the escape sequence for the font. For example, the HP LaserJet 14 point Helvetica bold font key contains this entry in the *Alias* field:

```
92286B \e(0U\e(s1p14v0s3b4T
```

Do not use the escape sequence that signifies font orientation (usually &10O or &11O). For example, if the escape sequence in the printer documentation reads

^{E}C&10O^{E}C(0U^{E}C(s1p14v0s3b4T

the entry in the *Alias* field would be

```
\e(0U\e(s1p6v0s0b4T
```

Notice that the printer documentation begins each escape sequence with ^{E}C. In the Font Tool, escape sequences always start with \e.

HP LaserJet Downloadable Fonts

When using downloadable fonts, special characters are used to precede the font file name. These characters indicate various attributes of the font file, as follows:

Backslash (\). Indicates a portrait font to download, if the page is printed in portrait orientation. For example if the name of the font file is HLV10R.SFL, the entry in the Alias field should be

```
\HLV10R.SFL
```

Minus Sign (–). Indicates a landscape font to download if the page is printed in landscape orientation.

Number Sign (#). Indicates a font to download regardless of the page orientation.

Plus Sign (+). If the font file can be printed in either landscape or portrait mode, the plus sign is used as a shortcut, rather than using both a backslash and minus sign. The plus sign is also included in the font file name. When the font is printed in portrait mode, the plus sign in the font file name changes to a P. When the font file is printed in landscape mode, the plus sign in the font file name changes to an L.

For example, if the name of the font file is HLV10R.SFL+, the entry in the Alias field should be

```
+HLV10R.SFL+
```

A font that uses a plus sign in the *Alias* field can print in both landscape and portrait mode.

Imagen Printers

Enter the name of the font as it is recognized by the printer. For example, 18 point Helvetica italic is called helvi18. So, the entry in the *Alias* field would be

```
helvi18
```

PostScript and LaserWriter+ Printers

The PostScript driver reads the *Alias* field to find the font names as recognized by PostScript printers. For example, the Helvetica bold font has this entry in the *Alias* field:

```
Helvetica-Bold
```

The driver also reads the Alias field for font cartridge names. Enter the name of the font as it is recognized by PostScript, enter a space, and then enter the font cartridge name.

Simple Printers

Printers using the simple device type use the *Alias* field in the same way as the daisy device printers.

Appendix C
Decimal/Hexadecimal Conversion Table

This appendix contains decimal and hexadecimal equivalents. The conversions in this appendix are helpful if your printer documentation uses decimal numbers instead of hexadecimal notation.

Table C-1. Decimal/Hexadecimal Conversion Table

0	00h	25	19h	50	32h	75	4Bh	100	64h
1	01h	26	1Ah	51	33h	76	4Ch	101	65h
2	02h	27	1Bh	52	34h	77	4Dh	102	66h
3	03h	28	1Ch	53	35h	78	4Eh	103	67h
4	04h	29	1Dh	54	36h	79	4Fh	104	68h
5	05h	30	1Eh	55	37h	80	50h	105	69h
6	06h	31	1Fh	56	38h	81	51h	106	6Ah
7	07h	32	20h	57	39h	82	52h	107	6Bh
8	08h	33	21h	58	3Ah	83	53h	108	6Ch
9	09h	34	22h	59	3Bh	84	54h	109	6Dh
10	0Ah	35	23h	60	3Ch	85	55h	110	6Eh
11	0Bh	36	24h	61	3Dh	86	56h	111	6Fh
12	0Ch	37	25h	62	3Eh	87	57h	112	70h
13	0Dh	38	26h	63	3Fh	88	58h	113	71h
14	0Eh	39	27h	64	40h	89	59h	114	72h
15	0Fh	40	28h	65	41h	90	5Ah	115	73h
16	10h	41	29h	66	42h	91	5Bh	116	74h
17	11h	42	2Ah	67	43h	92	5Ch	117	75h
18	12h	43	2Bh	68	44h	93	5Dh	118	76h
19	13h	44	2Ch	69	45h	94	5Eh	119	77h
20	14h	45	2Dh	70	46h	95	5Fh	120	78h
21	15h	46	2Eh	71	47h	96	60h	121	79h
22	16h	47	2Fh	72	48h	97	61h	122	7Ah
23	17h	48	30h	73	49h	98	62h	123	7Bh
24	18h	49	31h	74	4Ah	99	63h	124	7Ch

continued

Table C-1. Decimal/Hexadecimal Conversion Table (cont.)

125	7Dh	152	98h	179	B3h	206	CEh	233	E9h
126	7Eh	153	99h	180	B4h	207	CFh	234	EAh
127	7Fh	154	9Ah	181	B5h	208	D0h	235	EBh
128	80h	155	9Bh	182	B6h	209	D1h	236	ECh
129	81h	156	9Ch	183	B7h	210	D2h	237	EDh
130	82h	157	9Dh	184	B8h	211	D3h	238	EEh
131	83h	158	9Eh	185	B9h	212	D4h	238	EFh
132	84h	159	9Fh	186	BAh	213	D5h	239	F0h
133	85h	160	A0h	187	BBh	214	D6h	240	F1h
134	86h	161	A1h	188	BCh	215	D7h	241	F2h
135	87h	162	A2h	189	BDh	216	D8h	242	F3h
136	88h	163	A3h	190	BEh	217	D9h	243	F4h
137	89h	164	A4h	191	BFh	218	DAh	245	F5h
138	8Ah	165	A5h	192	C0h	219	DBh	246	F6h
139	8Bh	166	A6h	193	C1h	220	DCh	247	F7h
140	8Ch	167	A7h	194	C2h	221	DDh	248	F8h
141	8Dh	168	A8h	195	C3h	222	DEh	249	F9h
142	8Eh	169	A9h	196	C4h	223	DFh	250	FAh
143	8Fh	170	AAh	197	C5h	224	E0h	251	FBh
144	90h	171	ABh	198	C6h	225	E1h	252	FCh
145	91h	172	ACh	199	C7h	226	E2h	253	FDh
146	92h	173	ADh	200	C8h	227	E3h	254	FEh
147	93h	174	AEh	201	C9h	228	E4h	255	FFh
148	94h	175	AFh	202	CAh	229	E5h		
149	95h	176	B0h	203	CBh	230	E6h		
150	96h	177	B1h	204	CCh	231	E7h		
151	97h	178	B2h	205	CDh	232	E8h		

Appendix D
Translation Tables and Character Sets

This appendix lists the names of all the character sets and translation tables included in the standard editable font database. If you're not certain which character set or translation table to use, consult this appendix.

Translation Tables

AsciiToTextFont	Translates the standard ASCII character code set to the video character set. Used only for display purposes within the Font Tool.
Courier10	Maps the characters for the Courier10 print wheel.
Courier72	Maps the characters for the Courier72 print wheel.
CTGreektoAscii	Provides additional mappings to those supplied by the CTGreekToScientific12 translation table. (See below.)
CTGreekTo-Scientific12	Used to access the Greek font on the CT-2400 series printers.
CTtoAscii	Includes the standard ASCII character mappings, as well as the Document Designer character code set conventions. Used for the Simple devices and as a base for creating new translation tables that use the standard ASCII codes.
CtToCT2400	Translates the CTOS character code set to the character code set used by the CT-2400 series printers.

CtToEpsonL1	Translates the CTOS character code set to the Epson character code set. Used with Epson fonts.
CtToHpRoman8	Translates the CTOS character code set to the HP LaserJet character code set. Used with almost all HP LaserJet fonts.
CtToHpRoman8 NoUL	Used only with the 10 point Times bold font from the HP LaserJet B cartridge.
CtToImagenCour	Translates from the CTOS character code set. Used with Imagen Courier fonts.
CtToImagenGreek	Translates the CTOS character code set to Imagen codes, but uses the Greek alphabet instead of the Roman alphabet.
CtToImagenItalic	Translates the CTOS character code set to the Imagen codes. Used to shift the overstruck accents on Imagen italic fonts.
CtToImagenSymbol	Translates the CTOS character code set to the Imagen Symbol character code set. Used with Imagen symbol fonts.
CtToImagenText	Translates the CTOS character code set to the Imagen Text character code set. Used with all Imagen Roman and bold text fonts except Courier.
CtToPostScript-Dingbats	Translates the CTOS character code set to the PostScript Dingbats character code set. Used with the PostScript ITC Zapf Dingbats fonts. Honest.
CtToPostScript-Greek	Translates the CTOS character code set to PostScript codes, but uses the Greek alphabet instead of the Roman alphabet.
CtToPostScriptN	Translates the CTOS character code set to the PostScriptN character code set. Used with PostScript text fonts.
CtToPostScriptS	Translates the CTOS character code set to the PostScriptS character code set. Used with PostScript symbol fonts.

CtToVideo	Translates the CTOS character code set to the Screenfont Database character code set. Use this translation table to change raster fonts.
CtToVideoCT	Used in the Screenfont Database to access characters in the system font files when the raster font is 12 point.
CtToVideoSymbols	Used in the Screenfont Database to access characters in the various CTOS symbols font files, such as arrows and math symbols.
CubicPS96	Maps the characters for the CubicPS 96 print wheel.
ECS	Used by printers that use an extended character set (ECS). These printers commonly use 7-bit character codes instead of 8-bit character codes. In order to access the full 256 characters, these printers use the extended character set conventions.
EpsonL1toEpsonL2	Second level translation table used to translate the Epson character code set to Epson printer readable code.
EpsontoTextFont	Translates the Epson character code set to the TextFont character code set. Used only for display purposes within the Font Tool.
ImagenSymbol-ToGascii	Second level translation table used to translate the Imagen symbol character code set into printer-readable code.
ImagenText-ToGascii	Second level translation table used to translate the the Imagen text character code set into printer-readable code.
ImagenText-ToTextFont	Translates the Imagen text character code set to the TextFont character code set. Used only for display purposes within the Font Tool.

PostScriptNToNatl	Second level translation table used to translate the PostScript text font character code set to the PostScript printer character code set.
PostScript-ToTextFont	Translates the PostScript text character code set to the TextFont character code set. Used only for display purposes within the Font Tool.
Scientific10	Maps the characters for the Scientific10 print wheel.
VideoSymbolsToCt	Used only for display purposes within the Font Tool.
VideoToTextFont	Translates the raster font character code set to the TextFont character code set. Used only for display purposes within the Font Tool.

Character Sets

Ascii	Used by simple device types.
AsciiGreek	Used to access certain Greek characters on the CT-2400 series printers.
Courier10	Used by the Courier10 print wheel.
CT	The standard CTOS character code set.
CT-2400	Used by the CT-2400 series printers.
CT-2400Greek	Used to access the Greek font on the CT-2400 series printers.
CubicPS96	Used by the CubicPS96 print wheel.
Elite12	Used by the Elite12 print wheel.
Epson	Used by all Epson fonts.
HPRoman8	Used by almost all HP LaserJet fonts.
HPRoman8NoUL	Used only by the 10 point Times bold font from the HP LaserJet B cartridge.

ImagenCour	Used by all Imagen Courier fonts.
ImagenGreek	Used by all Imagen Greek fonts.
ImagenItalic	Used by all Imagen italic fonts.
ImagenSymbol	Used by all Imagen symbol fonts.
ImagenText	Used by all Imagen Roman and bold text fonts except Courier.
PostScriptN	Used by all PostScript text fonts.
PostScriptS	Used by all PostScript symbol fonts.
PSDingbats	Used by PostScript ITC Zapf Dingbats fonts.
PSGreek	Used by PostScript Greek fonts.
Scientific10	Used by the Scientific10 print wheel.
Video	Used by the Screenfont Database to display raster fonts in Document Designer.
VideoCT	Used by the Screenfont Database to display raster fonts in Document Designer.
VideoSymbols	Used by the Screenfont Database to display raster fonts in Document Designer.

Appendix E
Translation Table Escapes

This appendix lists the standard Font Database escapes. These escapes are used in translation tables for a variety of reasons, as explained below. Escapes in first level translation tables are used by the Font Tool; escapes in second level translation tables are used by the device driver.

00h (Nil)	Not used.
01h (Quote) (Used in first and second level translation tables.)	Indicates that the next hexadecimal code is a character, not an escape. Normally, hexadecimal codes below 20h are escapes. When preceded by 01h, however, these codes can be used as characters. For example, the Imagen character code for a bullet is 0Dh, which is below 20h, so it's entered in the translation table as 010Dh.
02h (16-bit Code) (Used only in second level translation tables.)	This escape indicates that the next two hexadecimal codes constitute a single 16-bit character code, instead of two 8-bit character codes. The first code is the high byte (bits 8-15). This escape is used in second level Imagen translation tables, since many Imagen characters can only be accessed through 16-bit codes resident in the printer. The 01 escape also functions as a Quote, in that the codes following it are always character codes, even if they are below 20h.

03h (Shift Left)
04h (Shift Right)
05h (Shift Up)
06h (Shift Down)
(Used only in first level translation tables.)

These escapes are used primarily to position overstruck accents. Overstruck accents are, by default, centered over the first character. With italic characters, the default placement is usually incorrect, and the accents need to be shifted to account for the italic slant. Other accents need to be shifted up so as to not print directly on top of the first character.

For example, the code for the Imagen italic capital A umlaut (Ä) reads:

41041205108A

This code uses both 04h and 05h. The first code, 41h, is the code for the A. The last code, 8Ah, is the code for the umlaut. The 0412h indicates to move the umlaut to the right 12 percent. The 0510h indicates to move the umlaut up 10 percent. (The percentages are in hexadecimal codes; 12 percent hexadecimal equals 18 percent decimal, and 10 percent hexadecimal equals 16 percent decimal.)

The actual distance to move the character is obtained by multiplying the percentage by the width of the first character in the hexadecimal string and dividing by 100. If you ever need to use these escapes, estimate the percentage and see how it prints.

07h (Alternate Character Set 1) (Used only in first level translation tables.)	Indicates that the next character code belongs to the first alternate character set. This escape is used primarily for overstriking characters where the overstrike character is in a different character set. For example, let's say you want to print an A umlaut (Ä), but the A is in the primary character set and the umlaut is in the first alternate character set. Assuming that the code for A is 41h and the code for the umlaut is 8A, the A umlaut code would read: 41078Ah The printer would print an A from the first charcter set, then overstrike the A with the umlaut from the first alternate character set.
08h (Alternate Character Set 2) (Used only in first level translation tables.)	Functions like 07h, but uses the second alternate character set.
09h (Alternate Character Set 3) (Used only in first level translation tables.)	Functions like 07h and 08h, but uses the third alternate character set.

0Eh (Shift-Out) (Used only in second level translation tables.)	Represents the standard ASCII shift-out character. This escape is used only by printers that use the extended character set (ECS). These printers use 7-bit character codes instead of 8-bit character codes, so they need to use the ECS to access characters outside of the standard 7-bit range. For example, the code for Ä is 0E200Fh. *0E*h is shift-out, 20h is the code for the character Ä, and *0F*h is shift-in.
0Fh (Shift-In) (Used only in second level translation tables.)	Represents the standard ASCII shift-in character. Used for the same purposes as a shift-out charcter, but it comes after the ECS character code, as shown in the example for shift-out.
10h (Use Width) (Used only in first level translation tables.)	Used when you want a character to have a different width than usual. For example, the character code for the leader dot is 2E105Fh, which instructs the printer to print the normal period character (2Eh) with the width of the underline character (5Fh). Thus, the dots are printed farther apart than usual.
1Bh (Escape) (Used only in second level translation tables.)	Represents the standard ASCII escape character. Used in second level translation tables that need to include the escape character in a code, for example the *EpsonL1toEpsonL2* table.

Appendix F
Font Database Character Set Conventions

This appendix explains the conventions that apply to the CTOS character code set, especially in terms of Document Designer and the Screenfont Database. Use this appendix when you create first level translation tables and modify screenfonts.

Required characters can be omitted from a primary translation table only if they will be picked up in an alternate translation table.

00h(Null)	Not used in any translation table.
08h(Required Backspace)	Required in screenfont translation tables. Not used in printer translation tables.
09h(Tab)	Required in screenfont translation tables. Not used in printer translation tables.
0Ah(Line Feed)	Required in screenfont translation tables. Not used in printer translation tables.
0Ch(Form Feed)	Required in screenfont translation tables. Not used in printer translation tables.
0Dh(Carriage Return)	Required in screenfont translation tables. Not used in printer translation tables.
14h (Nonbreaking Space)	Required in screenfont and printer translation tables. The translation must be to a blank character, usually the space character (20h), but other space characters can be used if you want a different width.

20h(Space)	Used in first level printer and screenfont translation tables. This character is the standard ASCII code for a space.
2Dh(Hyphen)	A line breaking character. Mapped to a minus sign in Document Designer, but can be changed to a true hyphen and still be a line breaking character.
80h (Nonbreaking Hyphen)	Not a line breaking character; mapped to a true hyphen.
81h (Discretionary Hyphen)	Not printed unless needed to break a line; mapped to a true hyphen.
82h (Discretionary Space)	Not printed, but is used in both printer and screenfont translation tables. This character should be mapped to the same code as the space character (usually 20).
83h–87h	Used for customization or nationalization.
88h(Leader Dot)	Required in both printer and screenfont translation tables. Always mapped to the leader dot character, usually 2E105F.
89h(Visible Paragraph Mark)	Not used in printer translation tables, but required in screenfont translation tables.
8Ah(Visible Heading Paragraph Mark)	Used with table of contents headings in the text of a document. Not used in printer translation tables, but required in screenfont translation tables.
8Bh(Visible Table of Contents Paragraph Mark)	Used for table of contents headings in an automatically generated table of contents. Not used in printer translation tables, but required in screenfont translation tables.

8Ch(Visible Outline Heading Paragraph Mark)	Not used in printer translation tables, but required in screenfont translation tables.
8Dh(Visible Null)	Not used in printer translation tables, but required in screenfont translation tables.
8Eh(Visible Space)	Not used in printer translation tables, but required in screenfont translation tables.
8Fh(Visible Nonbreaking Space)	Not used in printer translation tables, but required in screenfont translation tables.
90h(Visible Discretionary Space)	Not used in printer translation tables, but required in screenfont translation tables.
91h(Visible Nonbreaking Hyphen)	Not used in printer translation tables, but required in screenfont translation tables.
92h(Visible Discretionary Hyphen)	Not used in printer translation tables, but required in screenfont translation tables.
93h–9Fh	Reserved for future specialized use with Document Designer.
A0h–B5h	Primary range of characters used for nationalization. The CTOS character set includes foreign characters and other symbols.

B6h	Reserved for use with Document Designer.
B7h–BCh	Used for customization.
BDh	Reserved for use with Document Designer.
BEh–BFh	Used for customization.
C0h–FFh	Used for line and pattern drawing in the system screen font. Can also be used with some symbol fonts, for example, the HP LaserJet Linedraw font.

In addition to the printer and screenfont translation tables, there are also translation tables used only for display purposes within the Font Tool itself. These translation tables map the characters for the loadable font files *TextSet.font* and *VM003TextSet.font*. If you want to modify these translation tables or create new loadable font files, do not modify the codes used by the Font Tool to display text and forms; these reserved codes are 00h, 0Fh, 11h, 1Bh, 20h through 7Eh, C0h, C1h, C6h, CAh, CEh, DAh, DCh, E1h, E9h, EAh, EBh, and ECh.

Appendix G
Editable Font Database Format

The format of the Editable Font Database is upwardly compatible with future releases. The format of the Editable Font Database is documented so that OEMs and customers can write special applications to update the font data base. Blank space is reserved in most record types of the Font Database for future definition.

The format of the runtime Font Database, however, is tied to specific releases of the Font Database tool and the Font Service. The Font Generate command must be rerun whenever a new release of the Font Service is installed.

OEM applications can be written to add records to the editable data base. For example, an OEM application might be written to add relevant width tables and font keys from font or printer vendor tapes after implementing the corresponding device driver. All reserved fields in the editable data base records must be set to binary zeros in records which are added by OEM applications.

Alternatively, applications can update the font data base without using ISAM, by writing port files. Then the Font Database can be updated by invoking the Import command of the Font Tool. Each port file has the same name as the file being updated except that the suffix ".port" replaces the suffix ".isam" or ".data". Each of the port files is an RSAM file and has exactly the same record format as the ISAM or RSAM file that it updates.

The status field in each of the following records is a bit mask. The bits have the following meanings:

Mask	Description
01h	record has been edited by user
02h	record is referenced
04h	record is invalid
08h	record was software–generated
10h	reserved
20h	reserved
40h	reserved
80h	reserved

Records in which the software-generated bit is set and the user-edited bit is clear are marked in the Font Tool directories with a bullet (•). Other records are marked with a dagger (†) only if one of the other status bits are in an "interesting" state: Records in which the referenced bit is clear are marked with a dim dagger or bullet. Records in which the invalid bit is set are marked with a blinking dagger or bullet. Records created by OEM application software should have a status of 08h.

Fields named beginning with the letter f contain either binary 0 (**No**) or binary 1 (**Yes**). These fields in the reduction record may also contain the value of hex FE, the wild value which is denoted in the Font Tool by the asterisk (*).

Fields named beginning with 'sb' consist of a count byte followed by one fewer character bytes than the field size shown. Because ISAM does not recognize the sb data type, ISAM keys corresponding to the sb fields are defined as character fields beginning immediately after the count byte. For this reason, all sb fields must be padded to full size with trailing nulls (binary 0 bytes).

All fields beginning with 'wSize' contain font sizes in units of one twentieth (20) of a point. For example, 12 point is represented by the binary value 120. In the reduction record, the wild value for size fields (denoted in the Font Tool by an asterisk) is binary 0.

For more detailed information on the meaning and use of each field in the following records, please consult the corresponding sections on the use of the Font Tool.

Font Families File

FontFamily.data is an RSAM file. The record size is 29 bytes, subject to extension in future releases.

Offset	Size	Description
0	1	Reserved
1	1	status (bit flags)
2	25	sbFamily
27	1	fGenerate
28	1	fProportional
29	25	sbSimilarFamily

Pseudo Font Families File

FontPseudoFamily.isam is an ISAM file. The record size is 120 bytes. The key field (rgchFamily) is declared character:24.2.anu.w.

Offset	Size	Description
0	1	status (bit flags)
1	41	sbPseudoFamily
42	25	sbFamily
67	2	wSize
69	1	fBold
70	1	fItalic
71	25	sbDevice
96	24	reserved

Device Types File

FontDevice.data is an RSAM file. The record size is 35 bytes, subject to extension in future releases.

Offset	Size	Description
0	1	reserved
1	1	status (bit flags)
2	25	SbDevice
27	1	fGenerate
28	2	wHorizontalRes
30	2	wVerticalRes
32	1	fRaster
33	1	fPrint
34	1	fAlias
35	1	fScan

Font Key File

FontKey.isam is an ISAM file. The record length is 504 bytes. The key field is declared character:53.1.anu.w. Notice that the key field is a combination of logical fields (rgchDevice, rgchFamily, chBold, chItalic, rchPoints). So that ISAM sequencing will work, this field is entirely ASCII. Count bytes are omitted. RgchDevice and rgchFamily are padded with blanks (not nulls). Point size is represented as a two character numeral (range 1 to 99) with a blank in place of a leading zero. ChBold and chItalic are represented by characters 'Y' and 'N'. Device and family must be scanned backwards for the last non blank character in order to find the correct length of these strings.

Offset	Size	Description
0	1	status (bit flags)
1	24	rgchDevice
25	24	rgchFamily
49	1	chBold (Y or N)
50	1	chItalic (Y or N)
51	1	(must be a space)
52	2	rgchPoints (pic Z9)
54	2	wWidthNumerator
56	2	wWidthDenominator
58	1	fScaleByPoints?
59	25	sbWidthTable1
84	25	sbWidthTable2
109	25	sbWidthTable3
134	25	sbWidthTable4
159	13	sbCharacterSet1
172	13	sbCharacterSet2
185	13	sbCharacterSet3
198	13	sbCharacterSet4
211	41	sbAlias1
252	41	sbAlias2
293	41	sbAlias3
334	41	sbAlias4
375	25	sbRasterFont1
400	25	sbRasterFont2
425	25	sbRasterFont3
450	25	sbRasterFont4
475	29	reserved

Reduction Table File

FontReduction.data is an RSAM file. The record size is 174 bytes, subject to extension in future releases.

Offset	Size	Description
0	1	reserved
1	1	status (bit flags)
2	25	sbDomainFamily
27	2	wSizeDomainLo
29	2	wSizeDomainHi
31	1	fDomainBold
32	1	fDomainItalic
33	25	sbDomainDevice
58	25	sbRangeFamily
83	2	wSizeRange
85	1	fRangeBold
86	1	fRangeItalic
87	25	sbRangeDevice
112	1	reserved
113	61	sbComment

Character Sets File

FontCharacterSet.isam is an ISAM file. The record size is 248 bytes. The key field (rgchCharacterSet) is declared character:12.2.anu.w.

Offset	Size	Description
0	1	status (bit flags)
1	13	sbCharacterSet
14	2	reserved
16	25	sbTranslateFromCt
41	50	reserved
91	25	sbTranslateToPrinter
116	25	sbTranslateToVideo
141	25	sbNGenFontFile
166	31	sbComment
197	51	reserved

Width Tables File

FontWidth.isam is an ISAM file. The record size is 760 bytes. The key field (rgchWidthTable) is declared character:24.2.anu.w.

Offset	Size	Description
0	1	status (bit flags)
1	25	sbWidthTable
26	2	wMeasure
28	1	fInches
29	31	sbComment
60	13	sbCharacterSet
73	175	reserved
248	512	rgwWidth

Translation Tables File

FontTranslate.isam is an ISAM file. The record size is 2552 bytes. The key field (rgchTranslateTable) is declared character:24.2.anu.w. RgsbTranslate is an array of 256 sb's, each nine bytes long (one count byte and eight character bytes).

Offset	Size	Description
0	1	status (bit flags)
1	25	sbTranslateTable
26	31	sbComment
57	13	sbCharacterSet
70	13	sbCharacterSetTo
83	165	reserved
248	2304	rgsbTranslate

Raster Fonts File

FontRaster.isam is an ISAM file. The record size is 248 bytes. The key field (rgchRasterFont) is declared character:24.2.anu.w.

Offset	Size	Description
0	1	status (bit flags)
1	25	sbRasterFont
26	13	sbCharacterSet
39	25	sbWidthTable
64	2	wWidthNumerator
66	2	wWidthDenominator
68	180	reserved

Glossary

A

accents
Small symbols situated above or below certain characters that indicate a specific pronunciation of the character, for example, Ö and Å.

alias
In the Font Tool, a term used to designate any printer-specific escape sequences, font names, print wheel names, and so forth.

application
Any program or service used directly by an end user, for example, Document Designer.

ASCII code
American Standard Code for Information Interchange. Used to assign a set of basic punctuation and alphanumeric characters to a standard set of binary numbers, represented by their decimal or hexadecimal equivalents.

attribute
A characteristic that can be applied to a character or font, for example, bold or italic.

B

buffer
An area in memory used to store information temporarily.

C

character
Any letter, number, or symbol in your text.

character code set
A set of characters matched to a corresponding set of binary numbers.

character set
One or more related translation tables.

cluster
A number of workstations connected to a master workstation or shared resource processor (SRP) that share data, programs, and disk storage.

command line
The line on the Executive screen where you enter the name of the command you want to execute.

D

daisy printer
A character printer that uses interchangeable metal and plastic print wheels.

default value
The value that is assumed when no other value is specified.

device
Roughly synonymous with printer, but includes anything that outputs text, for example, monitors or plotters.

device driver
A system service that controls a printer and translates print jobs into the language of a specific printer. Device drivers are usually unique for each type of printer. The appropriate device driver must be installed at each workstation that has a printer attached.

distribution diskettes
The diskette(s) on which application programs are distributed.

dot matrix printer
See matrix printer.

downloadable fonts
Font files that are not resident in the printer but are instead stored on the hard disk. When you want to use these fonts, they are sent, or downloaded, to the printer.

E

editable font database
The storage facility for all font data. The editable font database is not accessed by any system service. When you edit your Font Database, you edit the editable font database.

escape sequences
Used by printers to access fonts or overstrike characters.

escapes
Hexadecimal codes used to assign different values to characters.

F

field
A line in a form or menu where you type in a value.

first level translation table
Used to translate the CTOS character code set to a printer character code set.

font
A collection of characters that have a distinct, uniform appearance. Sometimes a font is called a typeface.

font cartridge
Used by some printers to store the printer's font information. One printer can have many font cartridges that can be replaced at any time in order to access different fonts.

Font Database
A set of files that contain the information needed to format, display, or print documents in multiple fonts.

font family
The overall design of a font, for example Helvetica, Times, and Baby Teeth.

font key
Describes all important attributes of a font or points to where other attributes can be found. Every font in the Font Database has a font key.

Font Service
The system service that accesses the Font Database to provide fonts for client applications.

font substitution
Occurs when a printer can't print a requested font and another font is substituted.

Font Tool
An application program used to edit and organize data in the editable font database.

function key
The row of keys, marked **F1** to **F10**, at the top of the keyboard.

Function Key menu.
The Font Tool shows a Function Key menu as a row of blocks at the bottom of the screen that indicate the command assigned to each of the function keys.

G

Generic Print System (GPS)
A set of software programs that provide printing services for applications using the CTOS operating system. GPS manages all communications between your workstation and the printers attached to it.

H

hexadecimal codes
A numbering system based on 8-character binary numbers. Used extensively in the Font Tool.

K

keywords
Used by some device drivers to specify printer font selection criteria.

L

laser printer
A non-impact printer that produces high-resolution output.

ligature
A character made up of two or more connected characters. Ligatures are commonly used for combinations of f, l, and i. Ligatures are not included in the CTOS character code set, but they are included in the Screenfont character code set and some printer character code sets.

M

matrix printer
Mechanical printer that prints each character as a group of dots within a matrix rather than using a print wheel.

menu
A list of various command options from which you can make a choice.

monospaced font family
A font family in which all characters have the same width, for example, Courier.

N

nil value
A blank field in a reduction rule. Nil values handle unspecified values in a requested font.

nonbreaking hyphen
Used to keep text that contains hyphens together on a line.

O

optional hyphen
Used to hyphenate words at the end of a line. If at a later time text is edited and the break is no longer required at the end of the line, the optional hyphen doesn't appear.

original database
The editable font database installed by the GPS software diskettes.

overstrike
Overstriking occurs when a printer strikes more than once in the same place. This is usually done for accented characters or any characters that are comprised of separate parts. For example, certain printers print the O umlaut (Ö) by printing an O and then overstriking the character with an umlaut.

overwrite
To replace one file with another.

P

pipe drivers
Programs that translate ASCII or Generic Print Access Method (GPAM) files into a disk file in the language of the corresponding printer.

pitch
The width of a character (in characters per inch) of a font. Type that is 8 pitch has 8 characters per inch.

point
A measurement of 1/72 inch. Type height is measured in points.

port files
Used to transport information to and from the editable font database.

Print Manager
Application program used to interface with GPS in order to monitor and handle print requests.

print thimble
Used by thimble printers. Each character on a font is located on the print thimble, which prints the characters by striking a printer ribbon.

print wheel
A rotating disk (wheel) on a printer that presents characters at a single print position. Used with daisy printers.

printer independence
The ability to print any document on any printer, no matter what fonts are specified.

proportional font family
A font family in which all characters have different widths, for example, Times.

pseudo font family
Used to map entries in WP wheel set files to font keys.

R

Raster Font Designer
Application program used to create or modify raster fonts.

raster fonts
See screenfonts.

reduction rules
The rules that govern font substitution, as found in the Reduction Table.

Reduction Table
The part of the editable font database that lists the rules governing font substitution.

replacement font
The font that is substituted when the requested font can't be printed on the specified printer.

requested font
The font that you specify with the Fonts menu.

runtime font database
The part of the Font Database used by the Font Service to get font data for applications.

S

sans-serif type
A font that doesn't have any serifs, for example, Helvetica. *See also* serif type.

Screenfont Database
A superset of the Font Database that contains the Font Database *and* the raster font files.

screenfonts
The fonts displayed on a high-resolution monitor.

second level translation table
Used to translate a printer's character code set to Gascii or other printer-readable code.

serif type
A font that has short lines at the ends of the letters, for example, Times. *See also* sans-serif.

server
A workstation or shared resource processor (SRP) to which cluster workstations are connected. The master controls many system services, such as printing. Users can share the files and applications that are stored on the master.

spoke position
The location of a character on a print wheel.

standard font database
The editable or runtime font database installed when you install GPS. Some vendors may sell customized versions of the Font Database which differ from the standard Font Database.

status flags
Daggers or bullets that are displayed in the Font Tool to indicate a situation that requires action, for example, invalid or missing data.

T

translation table
Used to translate one character code set to another.

type style
Refers to the appearance of a font, for example, bold, italic, bold italic, and Roman (normal).

V

variant database
A set of files in the editable font database to which you modify or add font data. These files are kept separate from the original database.

W

wheel set files
Files used to store printer information for Word Processor, OFIS Writer, or pre OFIS Designer 2.0, and Document Designer 2.0 documents.

width table
Describes the width of each character in a proportional font.

wild cards
Asterisks found in reduction rules. Wild cards accept any value and leave any value unchanged.

WYSIWYG
(What-You-See-Is-What-You-Get.) On high-resolution monitors, application screenfonts match their corresponding printer fonts. In other words, when you specify 36 point Helvetica, the text on the screen is displayed in 36 point Helvetica.

Index

.data files, G-1
.font file, 4-6
.isam files, G-1
.port files, 4-2, 4-5, 4-7, G-1
1024x768_80Sys.font, 5-17
16-bit code escape, E-1

A

advance width, defined, 3-33
aliases, B-1 to B-5
  and downloadable fonts, B-4
  binary device driver, B-1
  CT-2400 device driver, B-1 to B-2
  daisy device driver, B-2
  Epson printers, B-3
  HP LaserJet printers, B-3
  Imagen printers, B-4
  LZ printers, B3
  PostScript printers, B-5
  simple printers, B-5
alternate character sets
  defined, 2-34 to 2-36
  escapes, E-3
  in font key, 3-31
ASCII, 2-23, 4-10, 4-11, 4-18
automatically created records, 3-6

B

baseline, defined, 3-32
Batch mode commands
  Create, 3-51
  Exit, 3-51
  Export, 3-52
  ExportAll, 3-52
  ExportAllVariant, 3-52
  ExportVariant, 3-52
  Generate, 3-53
  Import, 3-53
Batch mode commands, 3-51 to 3-54
  defined, 2-51
  using, 3-54
Binary device driver, B-1
bit map fonts *(See* raster fonts)
bit map width, defined, 3-33
bit maps, 4-1
Bitstream
  character ID, 4-7, 4-8, 4-11
  character set, 4-2, 4-10
blinking status flag, 3-7
bright status flag, 3-6
bullet status flag, 3-6

C

CCGI+ library, 4-1
cell height, defined, 3-32
character, defined, 1-3
character code sets
  CTOS, A-2 to A-13
    graphic representation of, A-2 to A-3
  defined, 2-17
  Screenfont, A-14 to A-20
character codes
  available for nationalization, 4-12 to 4-17

classes of
misc, 4-10
sym, 4-10
text, 4-10
CTOS, 4-10 to 4-11
defined, 2-17
description of, 4-9
downloaded to HP LaserJet printers, 4-12
character set conventions, F-1 to F-4
character sets
and translation tables, 3-25
available for nationalization, 4-9
Bitstream, 4-2, 4-7, 4-8
Character Sets directory, defined, 2-26 (*See also* Character Sets directory)
conventions, described, F-1 to F-4
00h, F-1
08h, F-1
09h, F-1
A0h to B5h, F-3
0Ah, F-1
0Ch, F-1
0Dh, F-1
2Dh, F-2
14h, F-1
20h, F-2
80h, F-2
81h, F-2
82h, F-2
83h to 87h, F-2
89h, F-2
90h, F-3
91h to 87h, F-3
92h, F-3
93h to 9Fh, F-3
8Ah, F-2
8Bh, F-2
8Ch, F-3
8Dh, F-3
8Eh, F-3
8Fh, F-3
creating, 3-24 to 3-26
CTOS, 4-2, 4-8, 4-9, 4-11
defined, 2-25
described, 2-25-2-27
example of creating, 5-11, 5-12
in standard editable font database, D-4 to D-5
naming, 3-24
PostScript, 4-2, 4-7 to 4-8
sample detail form, 2-26 to 2-27
when to add or modify, 2-25
working with, 3-23 to 3-36
Character Sets directory, defined, 2-26
Character translation tables *(See* Translation tables)
CODE-HELP key, 3-5
Commands
Batch mode commands, 2-51
Convert Wheel Set File, 2-50, 3-43 to 3-47
Create, 3-7, 3-51
Deinstall Font Service, 3-42
Delete, 4-4
Exit, 3-51
Export, 2-50, 3-48
batch mode, 3-52
example, 5-19
ExportAll, 3-52
ExportAllVariant, 3-52
Export Raster Fonts, 4-5, 4-8
ExportVariant, 3-49
batch mode, 3-52
Generate
batch mode, 3-53
Generate Runtime Database, 2-50, 3-41 to 3-43
Import, 2-50, 3-50, 4-4, 4-5
batch mode, 3-53, 5-15
example, 5-19
Install Font Service, 3-42

Invert, 3-23
Nationalize Scaling Fonts, 4-20 to 4-21
Propagate, 3-28, 4-18
Resize, 3-28
Restore, 3-43
Selective Backup, 3-43
Shuffle, 5-19
for translation tables, 3-23
for width tables, 3-28
Submit
FontDb, 3-9
ScreenfontDb, 3-9
Context Manager, 3-8
Convergent CT-2425, 2-16
Convergent CT-2450, 2-16
Convergent CT-2475, 2-16
Convergent LZ-5, 2-16
Convergent LZ-12, 2-17
Convergent *R12.font*, 2-38
Convert Wheel Set File command
defined, 2-50, 3-43 to 3-47
Copying runtime database, 3-42
Create command, 3-7, 3-51
CT-2400 device driver, B-1 to B-2
CT-2400 printer, device type for, 2-16
CTOS character code set, A-2 to A-13
graphic representation of, A-2 to A-3
cursor movement, 3-2
8Dh, F-3

D

dagger status flag, 3-6
Daisy device driver, B-2
Daisy device type, 2-16
decimal/hexadecimal conversion, C-1 to C-3
default device type
changing, 3-14
defined, 2-15
default font family
defined, 2-12
Deinstall Font Service command, 3-42
Delete command, 4-4
DELETE key, 3-5
Detail forms
accessing, 3-5
defined, 2-1
device driver
HP LaserJet, 4-1
PostScript, 4-12
Video120, 4-6, 4-21
device types
adding, 3-14 to 3-16
and downloadable fonts, 2-15
changing default device type, 3-14
default, 2-15 (*See also* Device Types directory)
defined, 2-13
described, 2-13 to 2-17
Device Types directory, defined, 2-14
in editable database, 2-16 to 2-17
in standard editable database
AP1302, 2-16
AP1303, 2-16
AP1305, 2-16
AP1307, 2-16
AP1314, 2-16
AP1324, 2-16
AP1327, 2-16
AP1329, 2-16
AP1351, 2-16
AP1351-1, 2-16
AP1354, 2-16
AP9206, 2-16
AP9208, 2-16
AP9215, 2-16
AP9215-1, 2-16
AP9230, 2-16

AP9415, 2-17
APDMatrix, 2-16
CT-2400, 2-16
EPFX286, 2-16
HP Download, 2-16
HP LaserJet, 2-16
HP PaintJet, 2-16
HP Plotter, 2-16
Imagen, 2-16
LaserWriter+, 2-16
LQ2250, 2-16
LZ-05, 2-16
LZ-12, 2-17
NEC3515, 2-16
PostScript, 2-17
Simple, 2-17
Video120, 2-17
VM-003, 2-17
modifying records, 3-13 to 3-14
supported, 2-16 to 2-17
when to add or modify, 2-14
working with, 3-13 to 3-16
Device Types directory
and downloadable fonts, 3-16
and plus sign status flags, 3-6
defined, 2-14
dim status flag, 3-7
Dir parameter, 3-16, 5-10
directories
accessing from main menu, 3-10
defined, 2-1
Document Designer, 4-1, 4-6
and font data, 2-50
and Font Database, 1-6
version 1.0, 2-39
downloadable fonts, 2-15
adding, 5-9
adding HP LaserJet, 5-11
Alias requirements, B-4
and Device Types directory, 3-16
and Print Manager, 3-16, 5-10
HP LaserJet, 5-9
Scan? field, described, 2-15
Draft fonts, 2-49

E

edit mode, 3-3
Editable Font Database, G-1 to G-8
defined, 1-4
files, G-3 to G-8
FontDevice.data, G-4
FontCharacterSet.isam, G-6
FontFamily.data, G-3
FontKey.isam, G-4 to G-5
FontPseudoFamily.isam, G-3
FontRaster.isam, G-8
FontReduction.data, G-6
FontTranslate.isam, G-7
FontWidth.isam, G-7
organization of, 2-1 to 2-2
entering the Font Tool, 3-8 to 3-9
Envision 420, 2-17
Envision 430, 2-17
Epson, device type for, 2-16
Epson FX-286, 2-16
Epson printers, B-3
escape codes
translation table, 4-8
Escapes
16-bit code, E-1
alternate character set 1, E-3
alternate character set 2, E-3
alternate character set 3, E-3
defined, 2-24 to 2-25
described, E-1 to E-4
00h, E-1
01h, E-1
02h, E-1
03h-06h, E-2
07h-09h, E-3
0Eh, E-4
0Fh, E-4
10h, E-4
1Bh, E-4
escape, E-4
first level, 2-25
in translation tables, 3-17
listed, E-1 to E-4

second level, 2-25
shift down, E-2
shift left, E-2
shift right, E-2
shift up, E-2
shift-in, E-4
shift-out, E-4
quote, E-1
use width, E-4
Executive
entering Font Tool, 3-8
Exit command, 3-51
Export command, 3-48
batch mode, 3-52
defined, 2-50
example, 5-19
Export Raster Fonts, 4-5, 4-8
Export Variant command, 3-49, 3-52
batch mode, 3-52
ExportAll command, 3-52
ExportAllVariant command, 3-52, 5-14
Exporting and importing font data, 3-47 to 3-51
Extended Multiplan, 2-49

F

file types
.data files, G-1
.isam files, G-1
.port files, G-1
First level translation table, 2-21 to 2-23
Font
scaling, 2-31
Font cartridges
adding, 5-4, 5-6 to 5-9
adding HP LaserJet, 5-7
and Alias field, 3-30
HP LaserJet, 1-2
Font data
and Document Designer, 2-51
and OFIS Designer, 2-51
description, 1-1
exporting and importing, 3-47 to 3-51
Font Database, 4-1, G-1 (*See also* Editable font database; Runtime font database; Variant database)
and Document Designer, 1-6 to 1-7
and OFIS Designer, 1-6 to 1-7
components, 1-10 to 1-11
description, 1-4 to 1-5
editing (overview), 1-8 to 1-10
Font families (*See also* Font Families directory)
adding, 3-12
default font family, 2-12
defined, 1-3, 2-11
described, 2-11 to 2-13
example of adding, 5-4
example of modifying, 5-3
Font Families directory, defined, 2-12
modifying records, 3-10 to 3-11
PostScript, 5-4
similar font families, 2-13
when to add or modify, 2-12
working with, 3-10 to 3-13
Font Families directory
and plus sign status flags, 3-6
defined, 2-12
described, 2-11
example of modifying, 5-3
Gen? field in, 2-12
Font files
1024x768_80Sys.font, 5-17
ConvergentR12.font, 2-38
modifying *TextSet.font*, F-4
modifying *VM003TextSet.font*, F-4

T1Sys.font, 5-17
TextSet.font, 2-26, 2-27
VM003TextSet.font, 2-27
VM003TextSet.font, 3-26
Font keys (*See also* Font Keys directory)
Alias, described, 2-33
and other font data, 2-30
and scalable fonts, 2-31
alternate character sets, 2-34
creating, 3-29 to 3-31
defined, 2-30
described, 2-30 to 2-36
example of creating, 5-9 to 5-10, 5-13 to 5-14
example of modifying, 5-4
Font Keys directory, defined, 2-31
modifying, 3-31
modifying HP LaserJet, 5-5
sample font key detail, 2-31 to 2-33
when to add or modify, 2-31
width factor, described, 2-32
working with, 3-29 to 3-31
Font Keys directory
defined, 2-31
font nationalization, 4-9 to 4-17
character codes available for, 4-9
characters available for, 4-13 to 4-17
translation tables
Bitstream, 4-11
PostScript, 4-12
Font Service
deinstalling, 3-41
installing, 3-42
Font substitution (*See also* Reduction rules)
and reduction rules, 3-39
defined, 1-5
Font terminology, 1-3
Font Tool
cursor movement in, 3-2
defined, 1-1
entering, 3-8 to 3-9
guidelines for using, 3-1 to 3-2
uses, 1-1 to 1-2
Font, defined, 1-3
FontCharacterSet.isam, G-6
FontDb Submit command, 3-9
FontDevice.data, G-4
FontFamily.data, G-3
FontKey.isam, G-4 to G-5
FontPseudoFamily.isam, G-3
FontRaster.isam, G-8
FontReduction.data, G-6
fonts
Imagen, 4-8
PostScript, 4-8, 4-9
shuffling order of, 4-8
FontTranslate.isam, G-7
FontWidth.isam, G-7
Format status line, 1-6
Function keys, 3-3 to 3-5
CODE-HELP, 3-5
COPY, 3-4
DELETE, 3-5
HELP, 3-5
INSERT, 3-4
MODIFY, 3-5
TOP, 3-3
REDO, 3-3
REVERT, 3-4

G

Gen? field
described, 2-12, 2-15
Generate Runtime Database command, 3-41 to 3-43
batch mode, 3-53
defined, 2-50
halting, 3-10
Generic Print System, 4-1, 4-2, 4-7, 4-11

H

HELP key, 3-5
Hexadecimal strings, 3-17
HP LaserJet
  adding downloadable fonts, 5-11
  adding font cartridge, 5-4, 5-6 to 5-9
  adding font keys, 5-8 to 5-9
  adding Linedraw font, 5-10 to 5-14
  Alias field, 5-8
  alias requirements, B-3
  creating width table, 5-6 to 5-7
  device type, 2-16
  downloadable fonts, 2-25, 5-10
  HP LaserJet+, 2-16
  HP LaserJet Series II, 2-16
  modifying font keys, 5-4, 5-5
  supported font cartridges, 1-2
HP PaintJet, 2-16
HP Plotter, 2-16

I

Imagen
  IP2, 2-16
  IP3, 2-16
  modifying hyphens, 5-16
Imagen fonts, 4-8
Imagen printers
  alias requirements, B-4
Import command, 3-50, 4-4, 4-5
  batch mode, 3-53, 5-15
  defined, 2-50
  example, 5-19
Insert key, 3-4
Install Font Service command, 3-42
invalid font keys, 3-41
invalid records, 3-7
Invert command, 3-23
invoking the Font Tool, 3-8 to 3-9

L

laser printers *(See* HP LaserJet; Imagen; PostScript)
LaserWriter, 2-16
LaserWriter+
  and scalable fonts, 2-31
  device type, 2-16
Left Offset, defined, 3-33
Linedraw font, adding, 5-12
LQ2250, 2-16
LZ device types, 2-16 to 2-17
LZ printers
  alias requirements, B3

M

Main menu, 2-10
  using, 3-9 to 3-10
Modify key, 3-5
modifying width tables
  after nationalization, 4-18
monospaced font family, defined, 1-3
moving records, 3-2

N

Nationalize Scaling Fonts command, 4-20 to 4-21
nationalization
  and system fonts, 5-16
  and variant databases, 2-7 to 2-9
  example, 5-20
  range of codes reserved for, 5-15
  shuffling translation tables, 5-19
nationalizing font outline files, 4-20 to 4-21
NEC3515, 2-16
Nil value reduction rules, 2-49

O

OFIS Designer, 4-1, 4-6
  and font data, 2-50
  and the Font Database, 1-6
  version 1.0, 2-39
original database
  defined, 2-2
  entering, 3-8
  modifying, 3-9
Outline status flag, 3-7

P

pitch, defined, 1-3
plus sign status flag, 3-6
point size, defined, 1-3
port files
  copying, 3-50
  exporting, 3-48
  exporting variant, 3-49
  importing, 3-50
PostScript
  adding font families, 5-4
  and scalable fonts, 2-31
  device types, 2-13, 2-17
PostScript character codes
  assigning differently, 4-12
  preassigned, 4-12
PostScript fonts, 4-8, 4-9
PostScript printers
  alias requirements, B-5
prescaled fonts
  commands related to, 4-4
  obtaining, 4-3 to 4-6
  using Export Raster Fonts, 4-3, 4-6
Print Manager
  and downloadable fonts, 3-16
print wheels
  creating translation tables for, 3-21, 3-22
printer independence, 1-6
printers (*See also* device types)
  adding, 2-9, 3-14
  AP1302, 2-16
  AP1303, 2-16
  AP1305, 2-16
  AP1307, 2-16
  AP1314, 2-16
  AP1324, 2-16
  AP1327, 2-16
  AP1329, 2-16
  AP1351, 2-16
  AP1351-1, 2-16
  AP1354, 2-16
  AP9206, 2-16
  AP9208, 2-16
  AP9215, 2-16, 2-17
  AP9215-1, 2-16
  AP9230, 2-16
  APDMatrix, 2-16
  Convergent CT-2425, 2-16
  Convergent CT-2450, 2-16
  Convergent CT-2475, 2-16
  Convergent LZ-5, 2-16
  Convergent LZ-12, 2-17
  Diablo 630 API-2, 2-16
  Diablo HPR05, 2-16
  Envision 420, 2-17
  Envision 430, 2-17
  Epson FX-286, 2-16
  HP LaserJet, 4-11, 4-12
    HP LaserJet+, 2-16
    HP LaserJet 500, 2-16
    HP LaserJet Series II, 2-16
  HP PaintJet, 2-16
  HP Plotter, 2-16
  Imagen
    IP2, 2-16
    IP3, 2-16
  LaserWriter, 2-16
  LaserWriter+, 2-16
  LaserWriter II, 4-1
  LQ2250, 2-16

NEC 3515, 2-16
PostScript, 4-12
Printronix MVP, 2-17
QMS, 2-17
QMS PS-800, 2-17
Qume Sprint 5, 2-16
Qume Sprint 9, 2-16
Qume Sprint 11/55 Plus, 2-16
TI 2108, 2-17
TI 2115, 2-17
Propagate command, 3-28
proportional font family, defined, 1-3
pseudo font families
defined, 2-39
described, 2-39 to 2-41
when to add, 2-40
Pseudo Font Families detail form
defined, 2-40
Pseudo Font Families directory
defined, 2-40

Q

QMS, 2-17
QMS PS-800, 2-17
quote escape, E-1

R

Raster Font Designer, 2-37
raster fonts
advance width, defined, 3-33
and screenfont database, 1-4
bit map width, defined, 3-33
caseline, defined, 3-32
cell height, defined, 3-32
defined, 2-36
described, 2-36 to 2-39
detail form, 3-34
Left Offset, defined, 3-33
Raster Fonts directory, defined, 2-37
Script form, 3-35
tuning, 3-32 to 3-37
variance, 3-35
when to modify, 2-37
guidelines, 3-38
width factor, 3-35
working with, 3-32 to 3-38 *(See also* Raster Fonts directory)
Raster Fonts directory, 4-4
defined, 2-37
record status flags (*See* Status flags)
record, defined, 2-2
REDO key, 3-3
reduction rules
adding, 3-40
and similar font families, 2-42
and type style, 2-45
defined, 2-41
described, 2-41 to 2-49
format, 2-41
how they are created, 2-46 to 2-47
modifying, 3-40
nil values, 2-49
order of, 2-46 to 2-49
proper rule order, 2-47 to 2-49
samples, 2-43 to 2-45
testing, 3-40
used to omit unused font keys, 2-41
when to add, 2-42
wild cards in, 2-45
working with, 3-39 to 3-40
Restore command, 3-43
Revert key, 3-4
Runtime font database, 4-3
bytes of buffer space, 3-42
copying, 3-42
defined, 1-5
distributing on floppy disks, 3-43
generating, 3-41 to 3-43
variant runtime database, 3-42

S

san serif, defined, 1-3
scalable fonts, 2-31
  in font keys, 3-30
Scaling Font Service
  character sizes
    CCGI+ library, 4-1
    Document Designer, 4-1
    OFIS Designer, 4-1
  components of, 4-2 to 4-3
    Export Raster Font Utility, 4-2 to 4-3
    Font Database *.port* files, 4-2
    font outline nationalization tool, 4-2
    master international font outline files, 4-2
    nationalized font outline files, 4-2
  configuring, 4-7 to 4-8
Screenfont character code set, A-14 to A-20
Screenfont database
  defined, 1-4
ScreenfontDb Submit command, 3-9
Screenfonts (*See* Raster fonts)
Script form, 3-35, 3-37
  changing, 3-38
second level translation tables, defined, 2-23 to 2-24
Selective Backup command, 3-43
serif, defined, 1-3
Shift down escape, E-2
Shift left escape, E-2
Shift right escape, E-2
Shift up escape, E-2
Shift-in escape, E-4
Shift-out escape, E-4
Shuffle command
  example of, 5-19
  for translation tables, 3-23
  for width tables, 3-28
shuffling font order, 4-8
similar font families
  and Font Families directory, 3-11, 3-12
  and reduction rules, 2-41
  and wheel set conversion, 3-47
  defined, 2-13
Simple printers
  alias requirements, B-5
soft fonts (*See* Downloadable fonts)
spacing pables (*See* Width tables)
starting the Font Tool, 3-8 to 3-9
status flags, 3-6 to 3-7
  blinking, 3-7
  bright, 3-6
  bullet, 3-6
  dagger, 3-6
  dim, 3-7
  outline, 3-7
  plus sign, 3-6
Submit commands
  FontDb, 3-9
  ScreenfontDb, 3-9
supported device types, 2-16 2-17
system fonts, 2-9, 5-17
  1024x768_80Sys.font, 5-17
  adding, 2-20
  T1Sys.font, 5-17

T

T1Sys.font, 5-17
TextSet.font, 2-27, 3-26
  modifying, F-4
thimble printers (*See* Daisy device type)
TI 2108, 2-17
TI 2115, 2-17
top key, 3-3
translation table escape codes, 4-8

translation tables (*See also* Translation Tables directory)
adding print wheels, 3-21
ASCII code, 2-23
character translation within, 2-22
creating, 3-19 to 3-22
defined, 2-17
described, 2-17 to 2-25
entering codes in, 3-20
escapes, 2-24 to 2-25
escapes, described, E-1 to E-4
00h, E-1
01h, E-1
02h, E-1
03h-06h, E-2
07h-09h, E-3
0Eh, E-4
0Fh, E-4
10h, E-4
1Bh, E-4
16-bit code, E-1
alternate character set 1, E-3
alternate character set 2, E-3
alternate character set 3, E-3
escape, E-4
quote, E-1
shift-down, E-2
shift-in, E-4
shift-left, E-2
shift-out, E-4
shift-right, E-2
shift-up, E-2
use width, E-4
example of creating, 5-11 to 5-12
example of modifying, 5-16
first level
defined, 2-21
sample, 2-21
for print wheels, 3-22
in standard editable font database, D-1 to D-4
Invert command, 3-23
level 1 (from CTOS), 4-7
level 2 (to printer), 4-7
modifying, 3-18
modifying Imagen hyphens, 5-16
naming, 3-19
rules regarding, 3-17 to 3-23
sample first level, 2-21 to 2-22
sample second level, 2-23 to 2-24
second level
defined, 2-21
sample, 2-23
Shuffle command, 3-23
shuffling, 5-19
Translation Tables directory, defined, 2-20
unmapped characters, 2-23
when to add or modify, 2-19 to 2-20
working with, 3-17 to 3-23
Translation Tables directory
defined, 2-20
tuning raster fonts, 3-32 to 3-37
guidelines, 3-38
type style
and reduction rules, 2-45
defined, 1-3
typefaces (*See* Font Families)

U

ungenerated records, 3-7
unused font data, excluding, 5-2
use width escape, E-4

V

Variant database, 2-2 to 2-9
advantages of using, 2-4 to 2-7
and batch mode commands, 3-52 to 3-53
changes in, 2-5
creating, 3-7

defined, 2-2
entering, 3-8
storing fonts in, 4-5
variant records, 3-6
Video120 device type, 2-17, 4-6, 4-21
View mode, 3-3
VM-003, 2-17
VM003TextSet.font, 2-27, 3-26
modifying, F-4

W

wheel set files
converting, 2-39, 2-50, 3-43 to 3-47
supported, 1-2
width tables
creating, 3-26 to 3-28
defined, 2-27
described, 2-27 to 2-30
entering widths in, 3-28
example of creating, 5-6 to 5-7
finding character widths, 3-26
Propagate command, 3-28
Resize command, 3-28
sample, 2-29 to 2-30
Shuffle command, 3-28
unit of measure within, 2-29
when to add, 2-28
working with, 3-26 to 3-28
Width Tables directory
defined, 2-28
wild cards, 2-45

NOTES

UNISYS

Help Us To Help You

Publication Title

Form Number

Unisys Corporation is interested in your comments and suggestions regarding this manual. We will use them to improve the quality of your Product Information. Please check type of suggestion:

☐ Addition ☐ Deletion ☐ Revision ☐ Error

Comments:

Name Telephone number ( )

Title Company

Address

City State Zip code

Cut along dotted line

Tape

Please Do Not Staple

Tape

Fold Here

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS MAIL PERMIT NO. 817 DETROIT, MI

POSTAGE WILL BE PAID BY ADDRESSEE

UNISYS CORPORATION
PRODUCT INFORMATION
MS 18-007
2700 NORTH FIRST STREET
SAN JOSE, CA 95134-2028